The Stochastic Economic Lot Scheduling Problem: Heavy Traffic Analysis of Dynamic Cyclic Policies

David M. Markowitz, Martin I. Reiman and
Lawrence M. Wein

#3863-95-MSA October, 1995

THE STOCHASTIC ECONOMIC LOT SCHEDULING PROBLEM: HEAVY TRAFFIC ANALYSIS OF DYNAMIC CYCLIC POLICIES

David M. Markowitz
Operations Research Center, M.I.T.

Martin I. Reiman
AT&T Bell Laboratories, Murray Hill

Lawrence M. Wein
Sloan School of Management, M.I.T.

We consider two queueing control problems that are stochastic versions of the economic lot scheduling problem: a single server processes N customer classes, and completed units enter a finished goods inventory that services exogenous customer demand. Unsatisfied demand is backordered, and each class has its own general service time distribution, renewal demand process, and holding and backordering cost rates. In the first problem, a setup cost is incurred when the server switches class, and the objective is to minimize the long run expected average costs of holding and backordering inventory and incurring setups. The setup cost is replaced by a setup time in the second problem, where the objective is to minimize average holding and backordering costs. In both problems we restrict ourselves to a class of dynamic cyclic policies, where idle periods and lot sizes are state-dependent, but the N classes must be served in a fixed sequence. Under standard heavy traffic conditions, these scheduling problems are approximated by diffusion control problems. The approximating setup cost problem is solved exactly, and the optimal dynamic lot sizing policy is found in closed form. Structural results and an algorithmic procedure are derived for the setup time problem. A computational study is undertaken to compare the proposed policy and several straw policies to the numerically computed optimal policy.

October 1995

We consider a queueing system scheduling problem that is motivated by a situation commonly found in make-to-stock manufacturing. The system consists of a single server, or machine, and multiple customer classes, which will be referred to as *products*. Each product has its own general service time distribution, and completed units are placed into a finished goods inventory; we assume that an ample amount of (costless) raw material inventory is available. Each product has its own renewal demand process that depletes the inventory, and unsatisfied demand is backordered.

We analyze two variants of the scheduling problem. In the *setup cost problem*, a cost is incurred when the machine switches production from one product to another; in the *setup time problem*, a random setup time is incurred when the server switches product. We restrict ourselves to the class of *dynamic cyclic policies*, where each product is serviced once per cycle and the order of production does not change. Thus, the server has three scheduling options at each point in time: Produce a unit of the product that is currently set up, change over to the next product in the cycle (and initiate service in the setup cost problem), or remain idle. Each product has its own costs per unit time for holding and backordering a unit in inventory. The objective in the setup time problem is to minimize the long run expected average inventory costs (that is, holding and backorder costs); the objective in the setup cost problem is to minimize the average inventory and setup costs.

These problems are prevalent in many industries because facilities that operate in a make-to-stock mode typically produce standardized products that require setups. The setup time problem is more realistic than the setup cost problem in most situations, but is also less amenable to analysis. The setup cost problem, however, is relevant for manufacturing systems that have embraced just-in-time (JIT) manufacturing and internalized their setup times; that is, they incur significant material, labor and/or capital costs to achieve essentially instantaneous switchovers.

This dynamic scheduling, or *lot-sizing*, problem is a stochastic version of the classic economic lot scheduling problem (ELSP), which is NP-hard (Hsu 1983) and has not been solved in general. Despite the vast literature devoted to the ELSP (see the survey paper by Elmaghraby 1978, and Zipkin 1991 for a list of more recent references), its deterministic viewpoint has probably prevented its widespread industrial use: The solution to

a deterministic problem in a make-to-stock setting will not hedge against uncertainty in future service times (e.g., machine failures) and demand, resulting in many costly backorders (see the numerical results in Federgruen and Katalan 1993).

Not surprisingly, the stochastic version of the ELSP appears to be analytically intractable. When the state space is taken to be discrete, the stochastic ELSP (or SELSP) can be viewed as a make-to-stock version of the dynamic scheduling problem for a *polling system*, which is a traditional (i.e., make-to-order) multiclass queue with setups. In fact, our paper can be viewed as a companion to Reiman and Wein (1994), who analyze the dynamic scheduling problem for a two-class polling system. The SELSP is more challenging than the polling scheduling problem, which also appears to defy exact analysis, because of the nonlinear cost structure and the lack of an imposed boundary at the origin. Despite its difficulty, this problem has been the subject of a recent flurry of activity. Graves (1980) develops a Markov decision model for a one-product problem, and uses it to develop a heuristic for the SELSP in a periodic review setting. Leachman and Gascon (1988), Gallego (1990) and Bourland and Yano (1995) develop heuristic lot-sizing algorithms for the ELSP with stochastic demands that are rooted in the solution to the deterministic ELSP; the first of these papers considers a discrete time problem with nonstationary demand. Sharifnia, Caramanis and Gershwin (1991) employ a hierarchical approach to develop heuristic policies for a stochastic fluid version of the problem. Federgruen and Katalan (1993, 1994) develop accurate distributional approximations for polling systems, and use these to analyze the performance of a class of periodic base stock policies for the SELSP. Anupindi and Tayur (1994) also consider a class of periodic base stock policies, and use a simulation based approach (infinitesimal perturbation analysis and gradient search) to obtain good base stock policies for a variety of performance measures. Sox and Muckstadt (1995) formulate the SELSP as a stochastic program and propose a heuristic decomposition algorithm to solve it. Qiu and Loulou (1995) formulate the problem as a semi-Markov decision process, and numerically compute the optimal solution in the two-product case; this is the only paper to date to gain any insight into the nature of the optimal solution to the SELSP.

As in Reiman and Wein, we employ heavy traffic approximations in an attempt to

make further progress with this problem. This approach assumes that the server must be busy the great majority of time in order to meet average demand over the long term. We draw upon the results of Coffman, Puhalskii and Reiman (1995a, 1995b) to approximate the setup cost and setup time problems by diffusion control problems. Their results allow the analysis of the problems to separate onto two different time scales. On the time scale over which the total workload content of the inventory varies, we solve a one-dimensional diffusion control problem to determine the machine idling policy; this time scale corresponds to the standard heavy traffic normalization, where time and space are compressed by the factors n and $\sqrt{n}$, respectively. Starting with this normalization, we expand time by a factor of $\sqrt{n}$, so that both time and space are compressed by $\sqrt{n}$; on this time scale, the total workload content of the inventory is fixed and the individual inventory levels vary according to a fluid model. Here, deterministic optimization is used to derive the state-dependent lot sizes when the server is busy. These two controls are decoupled in the setup cost problem, and a closed form solution is derived. In the setup time problem, however, the lot sizes derived under the fluid scaling impact the drift of the one-dimensional diffusion process that arises under the standard heavy traffic normalization. In this case, we develop some structural results about the optimal solution and provide an algorithmic procedure to solve the one-dimensional diffusion control problem. Although no weak convergence results are provided to rigorously justify our analysis, we have no doubt that our results are essentially correct; consequently, we conjecture that our proposed policies are asymptotically (in the heavy traffic limit) optimal for the two-product problems, where the optimal policies *are* dynamic cyclic policies.

In an attempt to both assess the effectiveness of some simpler policies and synthesize some of the existing literature on the SELSP, we perform a heavy traffic analysis of two straw policies that are closely related to those considered by Federgruen and Katalan (1993) and by Sharifnia, Caramanis and Gershwin. Then a computational study is undertaken that compares our proposed policies and these two straw policies to the numerically derived optimal policy for a variety of two-product problems; several five-product problems are also examined.

The explicitness of our results reveals a number of new and unexpected insights into

the nature of the optimal solution to the SELSP. Readers who are not curious about the mathematical details but who wish to obtain a deeper understanding of the SELSP may find it useful to bypass the heavy traffic analysis and focus on §1.8, §2.5 and §3.4, where the key insights and observations are collected.

The remainder of the paper is divided into four sections. §1 and §2 are devoted to the analysis of the setup cost and setup time problems, respectively. The computational study is described in §3 and concluding remarks are made in §4.

1. THE SETUP COST PROBLEM

1.1. Problem Description. A single server, or machine, produces N types of products. Each product $i = 1, 2, \ldots, N$ has its own general service time distribution with service rate μ_i and coefficient of variation (standard deviation divided by the mean) c_{si}. The demand for each product follows a renewal process, where the mean and coefficient of variation of the interdemand times are given by λ_i^{-1} and c_{di}, respectively. Our results easily generalize to correlated compound renewal processes; see Reiman (1984) for details. Hence, the traffic intensity, or long run average server utilization, is $\rho = \sum_{i=1}^{N}(\lambda_i/\mu_i)$ and $\rho_i = \lambda_i/\mu_i$ is the utilization for product i.

Let $\tilde{I}_i(t)$ denote the number of units of product i in inventory at time t. A service completion of product i at time t increases $\tilde{I}_i(t)$ by one, and a unit of product i demanded at time t depletes $\tilde{I}_i(t)$ by one. If a demand for a unit of product i occurs when $\tilde{I}_i(t) \leq 0$, we say that this unit of demand is backordered.

Since the scheduler follows a dynamic cyclic policy, only three options are available at each point in time: Produce the product that is currently set up, switch over and initiate production of the next product in the cycle, or sit idle. Because a setup is costly and instantaneous, the option of switching to another product and then idling is clearly suboptimal and will not be considered. The server is assumed to work in a preemptive-resume manner, although the subsequent heavy traffic analysis is too crude to distinguish between this and the non-preemptive discipline.

A cost $\bar{h}_i$ is incurred per unit time for holding a unit of product i in inventory, and a cost $\bar{b}_i$ is incurred per unit time for backordering a unit of product i. Let us also define the

cost indices $h_i = \bar{h}_i\mu_i$ and $b_i = \bar{b}_i\mu_i$, which represent holding and backorder cost rates per unit of expected work in inventory. These indices play a key role in our analysis, and are the analog to the classic "$c\mu$" index in stochastic scheduling theory (e.g., Cox and Smith 1961). Without loss of generality, the products are numbered so that $h_N = \min_{1\le i\le N} h_i$. For notational convenience, we assume that $b_N = \min_{1\le i\le N} b_i$, so that the product with the smallest holding cost index also has the smallest backorder cost index; we show later that this restriction does not change the analysis from that of the more general case. Consequently, product N will often be referred to as the *least cost product*.

Because products are produced once per cycle in a fixed sequence, the performance of the system depends on the setup costs only through the total setup cost per cycle, a quantity we denote by K. Of course, the optimal sequence of products can be found by solving a travelling salesman problem, where the "distance" between "cities" i and j is given by the setup cost incurred when switching from product i to j. For notational convenience, we simply assume that a setup cost K/N is imposed whenever the server switches from one product to another.

The scheduling policy determines the inventory process $\tilde{I}_i$ and the counting process J, where $J(t)$ denotes the cumulative number of setups incurred by the server up to time t. Although the scheduling policy, and hence the stochastic processes $\tilde{I}_i$ and J, can be rigorously defined in terms of a sequence of starting and stopping times for each product, we omit this detailed specification because it is not required in our subsequent analysis. Our problem is to find a scheduling policy, which is nonanticipating with respect to $\tilde{I}_i$, to minimize

$$\limsup_{T\to\infty} \frac{1}{T} E\Big[\int_0^T \sum_{i=1}^N (\bar{h}_i \tilde{I}_i^+(t) + \bar{b}_i \tilde{I}_i^-(t))dt + \frac{K}{N} J(T)\Big], \tag{1}$$

where $x^+ = \max(x, 0)$ and $x^- = \min(x, 0)$.

1.2. The Heavy Traffic Normalizations. The heavy traffic averaging principle described in §1.3 allows us to compactly characterize the class of dynamic cyclic policies in heavy traffic. Since we will be optimizing over this class of policies, there is no need to precisely formulate the (more general) control problem that approximates (1) in heavy traffic, and it will suffice to describe the heavy traffic conditions and normalizations.

The standard heavy traffic condition states that the server must be busy the majority of time to meet average demand. To prove a heavy traffic limit theorem, one typically defines a sequence of control problems indexed by n, where the traffic intensity of the n^{th} system approaches unity. Because a limit theorem will not be proved here, we ease the notational burden by assuming the existence of a large integer n satisfying $\sqrt{n}(1-\rho) = c$, where c is positive and $O(1)$. Our recommended policy will turn out to be independent of the heavy traffic scaling parameter n. Let the process $\tilde{W}_i$ denote the workload content embedded in the finished goods inventory $\tilde{I}_i$. If $\tilde{I}_i(t)$ is positive then $\tilde{W}_i(t)$ represents the sum of the service times for units in inventory, and if $\tilde{I}_i(t)$ is negative then $\tilde{W}_i(t)$ represents the amount of time that a continuously busy server requires to clear product i's backlog. The process $\tilde{W}_i$ will be referred to as the *workload* process for product i. The standard heavy traffic scalings will be employed to define the normalized inventory process $I_i(t) = \tilde{I}_i(nt)/\sqrt{n}$ and the normalized workload process $W_i(t) = \tilde{W}_i(nt)/\sqrt{n}$ (throughout the remainder of the paper, a "tilde" denotes the unscaled version of the normalized process). Although the workload process $\tilde{W}_i$ is not observable by the scheduler when inventory is backordered, the normalized workload process W_i is more convenient for analysis than the normalized inventory process I_i. However, the linear identity $I_i = \mu_i W_i$, which is known to hold for a wide variety of queueing systems in heavy traffic, will be employed for translating the solution of the heavy traffic control problem into a scheduling policy that is expressed in terms of the unscaled inventory process $(\tilde{I}_1, \ldots, \tilde{I}_N)$.

The objective function (1) consists of inventory costs and setup costs. To perform a nontrivial heavy traffic analysis, we need these two costs to be of the same order of magnitude in the approximating control problem. Since the inventory contents and time are normalized in the heavy traffic control problem, the relative magnitude of the inventory costs and setup costs also need to be scaled in order to achieve the aforementioned balance. The appropriate normalization (see Reiman and Wein for details) is to reduce the setup cost K by a factor of n relative to the inventory costs, and consequently we leave the inventory costs $\bar{h}_i$ and $\bar{b}_i$ unscaled at $O(1)$, and define the normalized setup cost $k = K/n$. That is, if the setup cost is smaller than $O(n)$, then setup costs will be negligible in heavy traffic, and if they are larger than $O(n)$, then they will dominate the inventory costs in

heavy traffic. Thus, heavy traffic conditions for the setup cost problem require the server to be busy most of the time, and require the setup cost to be very large.

1.3. A Preliminary Heavy Traffic Result. A recent heavy traffic result obtained by Coffman, Puhalskii and Reiman (1995a) provides the basis for our analysis. This result, which will be referred to as the CPR result, considers a traditional multiclass (the result is only proved for the $N = 2$ case, and is conjectured for $N > 2$), single server queueing system with renewal arrival processes, general service time distributions and no setup times. A cyclic exhaustive polling mechanism is employed: The server serves each class to exhaustion and then switches to the next class. Let us reuse the service time notation μ_i and c_{si}, and allow the demand parameters λ_i and c_{di} in the SELSP to also characterize the arrival process to this queueing system. If we let V_i denote the normalized workload for class i in the queue, then limit theorems in Iglehart and Whitt (1970) and Reiman (1988) imply that the total workload process $V = \sum_{i=1}^{N} V_i$ is well approximated under heavy traffic conditions by a reflected Brownian motion on $[0, \infty)$ with drift $-c$ and variance

$$\sigma^2 = \sum_{i=1}^{N} \frac{\lambda_i}{\mu_i^2}(c_{di}^2 + c_{si}^2). \tag{2}$$

See Harrison (1985) for a definition of this process, which we denote by RBM $(-c, \sigma^2)$.

The CPR result allows us to obtain information about the multidimensional workload process. It provides a *time scale decomposition* for this system: On the time scale giving rise to RBM for the total workload, the N-dimensional workload $(V_1, \ldots, V_N)$ process moves infinitely quickly (in the asymptotic limit) along a constant workload, piecewise-linear path connecting points where the server exhausts a product. In the two-product case, the path is the line segment from $(0, V)$ to $(V, 0)$; in the case of three identical products (with the same service and demand rates), the path consists of the line segments connecting the points $(0, V/3, 2V/3)$, $(V/3, 2V/3, 0)$ and $(2V/3, 0, V/3)$.

We make the crucial assumption that the time scale decomposition holds not just for the cyclic exhaustive policy (which corresponds to a cyclic base stock policy in the SELSP problem), but for more general cyclic policies; this assumption should not raise any objections because the time scale decomposition is *inherent in the heavy traffic scaling*.

The time scale decomposition gives rise to an *averaging principle* that facilitates our analysis in the following way: If we consider the heavy traffic normalization and then expand the time scale (i.e., slow down time) by a factor of $\sqrt{n}$, then the total workload stays constant and the N-dimensional workload behaves as a fluid (i.e., moves at a finite rate in a deterministic fashion).

1.4. An Overview of the Analysis. It is useful to view the control policy as consisting of two interrelated decisions: A busy/idle policy and a dynamic lot-sizing policy that specifies what the server should do while working. We begin by characterizing the busy/idle policy. The CPR result and the well known relationship between queueing systems and production/inventory systems (e.g., Morse 1958) imply that the system state of the heavy traffic control problem is the one-dimensional total workload process $W = \sum_{i=1}^{N} W_i$, which measures the total machine time embodied in the current finished goods inventory. Furthermore, since setup times are zero, the total workload process is only affected by the server's busy/idle policy, not by how often the server switches among products. Hence, the only reasonable form of the optimal busy/idle policy is for the server to stay busy if $W(t) < w_0$ and to idle if $W(t) \geq w_0$, for the unspecified control parameter w_0. The quantity w_0 will often be referred to as the *idling threshold*, and can be viewed as an *aggregate base stock level*. The CPR result and the one-to-one relationship between queueing systems and production/inventory systems (i.e., $W(t) = w_0 - V(t)$) imply that the total workload process W is a RBM (c, σ^2) on $(-\infty, w_0]$ under this busy/idle policy.

Although the lot-sizing policy does not influence the total workload W, it does affect the rate at which inventory costs and setup costs are incurred when $W(t) = w$. Therefore, a two-step procedure, with each step being performed at a different time scale, can be used to find the optimal dynamic cyclic policy. In the first step, CPR's averaging principle is used to find the lot-sizing policy that minimizes the average (inventory plus setup) cost incurred as the individual inventory levels oscillate deterministically while the total workload remains constant at w; let us call the resulting minimum cost $c(w)$. By solving a family of deterministic optimization problems indexed by the total workload w, we are able to construct a dynamic (i.e., workload-dependent) lot-sizing policy. In the second

step, we find the aggregate base stock level w_0 that minimizes the long run average cost

$$\int_{-\infty}^{w_0} c(w)\frac{2c}{\sigma^2}e^{-2c(w_0-w)/\sigma^2}\,dw, \tag{3}$$

where we have used the fact (e.g., Harrison) that RBM $(-c,\sigma^2)$ on $[0,\infty)$ has an exponential stationary distribution with parameter $2|c|/\sigma^2$.

In §1.5, we calculate the average cost incurred by a generic dynamic cyclic policy when the total workload equals w. The optimal heavy traffic cyclic policy is found in §1.6. Finally, the heavy traffic normalizations are reversed in §1.7 to obtain the proposed scheduling policy as a function of the unscaled inventory levels and the server location.

1.5. Construction of Dynamic Cyclic Policies. The goal of this subsection is to find the cost associated with any dynamic cyclic policy when the total workload $W(t) = w$. The CPR result makes this calculation tractable: Starting with the heavy traffic normalization $W_i(t) = \tilde{W}_i(nt)/\sqrt{n}$, we slow down time by a factor of $\sqrt{n}$ to obtain a fluid scaling, $\bar{W}_i(t) = \tilde{W}_i(\sqrt{n}t)/\sqrt{n}$. At this time scaling, the process $W(t) = \tilde{W}(nt)/\sqrt{n}$ is fixed at the value w, and the $N-$ dimensional workload $\bar{W}_i$ moves at a finite rate in a deterministic manner. Because many fluid cycles occur before the diffusion process W changes value, it suffices to analyze the deterministic behavior of one cycle.

A cyclic policy is essentially characterized by the lot size for each product (or equivalently, since the analysis is deterministic, the length of time each product is served in a cycle). Because idleness is only incurred when the total workload reaches a certain base stock level, we assume that no idleness is incurred during a cycle. When $W(t) = w$, a cyclic policy is best viewed as a closed $N-$dimensional deterministic path in the constant workload hyperplane $\sum_{i=1}^{N} w_i = w$; that is, the process traverses the same path repeatedly, once per cycle.

Although a cyclic policy can be specified in many ways, we choose a particular characterization that is convenient for analysis. A cyclic policy (or, equivalently, the closed loop generated by the policy) will be defined by $N+1$ quantities: The *cycle length* τ and the *cycle center* $x^c = (x_1^c, \ldots, x_N^c)$. These control parameters are actually functions of the total workload w, but this dependence will be suppressed for improved readability.

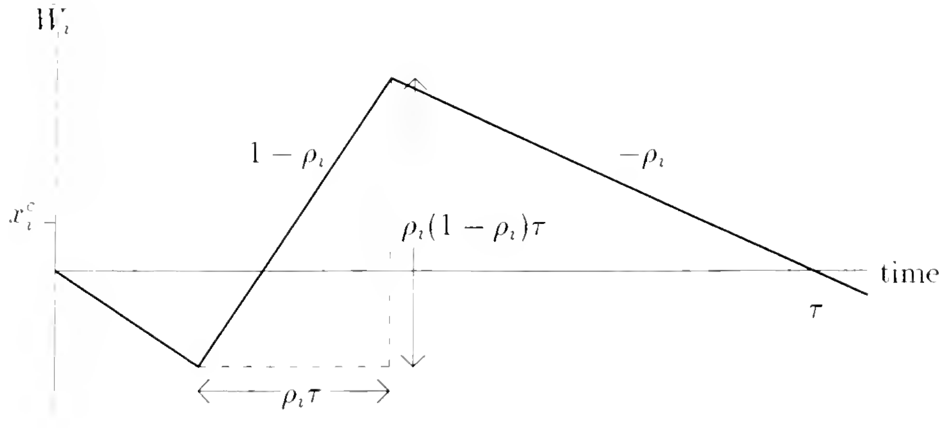

Figure 1: Workload fluctuation over a cycle.

The cycle length τ is the length of time required to perform a cycle, and x_i^c is product i's "center of fluctuation", or equivalently, the average amount of this product's inventory over the course of a cycle. Because the transient effects associated with initiating or temporarily moving a cycle vanish in the heavy traffic time scaling, the cycle center x^c can be placed anywhere in the constant workload hyperplane.

We begin by examining the deterministic behavior of the individual product workload levels W_i under a cyclic policy when $W(t) = w$. For the system to remain balanced, the amount of each product produced per cycle must equal the amount demanded, and hence each product must be produced a fraction ρ_i of the time; we assume that ρ equals one throughout this fluid analysis, so that the server is busy throughout the cycle. Thus, for an arbitrary instantaneous total workload w and cycle time τ, each product i must be serviced for $\rho_i\tau$ units of time per cycle. Therefore, when the machine is servicing product i, the work content in this product's inventory is depleted at rate ρ_i and is replenished at rate one, and W_i increases at the fixed rate $1-\rho_i$ for $\rho_i\tau$ units of time per cycle. For the remaining $(1-\rho_i)\tau$ time units in the cycle when product i is not being produced, the workload inventory is decreasing at rate ρ_i. To uniquely determine the behavior of a cyclic policy, a reference starting point also needs to be specified. We use x_i^c, product i's average inventory level, as the reference point. Readers are referred to Figure 1 for a

reinforcement of these notions.

The cost of a cyclic policy can be expressed in terms of the cycle length τ, the cycle center x^C (an N–dimensional vector) and the total workload w. To find the average inventory cost for product i per unit time, $c_i(\tau, x_i^c, w)$, we integrate the inventory cost over the cycle and divide by the cycle length. The average cost breaks down into three regions depending on whether the product is entirely held throughout the cycle, is held and backordered or is entirely backordered. Figure 1 allows us to deduce that

$$c_i(\tau, x_i^c, w) = \begin{cases} h_i x_i^c & \text{if } x_i^c > \frac{\tau\rho_i(1-\rho_i)}{2} \\ (b_i + h_i)\frac{\tau\rho_i(1-\rho_i)}{8} + \frac{b_i+h_i}{2\tau\rho_i(1-\rho_i)}(x_i^c)^2 + \frac{h_i-b_i}{2}x_i^c & \text{if } 0 \in \left[x_i^c \pm \frac{\tau\rho_i(1-\rho_i)}{2}\right] \\ -b_i x_i^c & \text{if } x_i^c < -\frac{\tau\rho_i(1-\rho_i)}{2} \end{cases} . \tag{4}$$

The total average cost, $c(\tau, x^c, w)$, which includes inventory and setup costs, is

$$c(\tau, x^c, w) = \sum_{i=1}^{N} c_i(\tau, x_i^c, w) + \frac{k}{\tau}. \tag{5}$$

1.6. The Optimal Dynamic Cyclic Policy. Our cyclic policy consists of three controls: The aggregate base stock level w_0 and, for each total workload level w, the cycle length τ and the N–dimensional cycle center x^c. In this subsection, the optimal dynamic cyclic policy is derived in three stages: (i) find the optimal cycle center x^c in terms of arbitrary w and τ, (ii) optimize over the cycle time τ in terms of an arbitrary w, and (iii) substitute the derived cost function $c(w)$ into equation (3) and find the optimal idling threshold w_0.

The Optimal Cycle Center. We begin by showing the existence of a cost-minimizing cycle center x^c for a given total workload w and cycle length τ. Note that the cost function $c(\tau, x^c, w)$ is differentiable with respect to x^c and its derivative is continuous. If one ignores the constant workload constraint $\sum_{i=1}^{N} x_i^c = w$, for fixed τ the cost function in terms of x^c is piecewise-quadratic with linear edges; its second derivative is a nonnegative step function. Thus $c(\tau, x^c, w)$ is convex and the restriction of the cost function to the constant workload hyperplane determined by w is also convex. This fact implies the

existence of a solution to the constrained minimization problem: Choose x^c to minimize $c(\tau, x^c, w)$ subject to $\sum_{i=1}^N x_i^c = w$.

Now we use the cost function in (4) to find the optimal cycle center. The total workload constraint can be used to eliminate one variable and express the cost function as a piecewise polynomial function of $N-1$ variables. Any $N-1$ components of x^c can be used. Over the constant workload hyperplane, the polynomial order of the $N-1$ variables fluctuates between one and two depending on whether $|x_i^c| > \frac{\tau\rho_i(1-\rho_i)}{2}$ or $|x_i^c| \leq \frac{\tau\rho_i(1-\rho_i)}{2}$, respectively. For the gradient to be equal to zero, each of the $N-1$ variables must be quadratic. Consequently, at the optimal x^c, at least $N-1$ of the $c_i(\tau, x_i^c, w)$'s are of order two, with the remaining component possibly being linear. To see this, suppose that some of $c_i(\tau, x_i^c, w)$'s are not of order two, and let j denote the index of such a term. If we eliminate x_j, the gradient equation can then be written as

$$\nabla_{x^c}\left[\sum_{\substack{i=1\\ i\neq j}}^{N} c_i(\tau, x_i^c, w) + c_j\left(\tau, w - \sum_{\substack{i=1\\ i\neq j}}^{N} x_i^c, w\right)\right] = 0 \ .$$

This equation will have a solution only if the remaining $N-1$ $c_i(\tau, x_i^c, w)$'s are quadratic.

The following proposition greatly simplifies our analysis.

Proposition 1 *If there are only $N-1$ quadratic $c_i(\tau, x_i^c, w)$ terms in the total cost function at the optimal x^c, then the linear term must be $c_N(\tau, x_N^c, w)$.*

Proof: This fact is most easily seen by examining the function $c_i(\tau, x_i^c, w) + c_N(\tau, x_N^c, w)$, where c_i is linear and c_N is quadratic. By (4), the sum is

$$h_i(x_i^c)^+ + b_i(x_i^c)^- + (b_N + h_N)\frac{\tau\rho_N(1-\rho_N)}{8} + \frac{b_N + h_N}{2\tau\rho_N(1-\rho_N)}(x_N^c)^2 + \frac{h_N - b_N}{2}x_N^c. \quad (6)$$

The tradeoff between x_i^c and x_N^c can be examined by looking at this sum along the line $x_i^c + x_N^c = w'$ (with w' arbitrary). Substituting $w' - x_N^c$ for x_i^c into equation (6) and taking the derivative with respect to x_N^c yields

$$-h_i + \frac{h_N - b_N}{2} + \frac{b_N + h_N}{\tau\rho_N(1-\rho_N)}x_N^c \quad \text{if } x_N^c > w', \quad (7)$$

$$b_i + \frac{h_N - b_N}{2} + \frac{b_N + h_N}{\tau\rho_N(1-\rho_N)} x_N^c \quad \text{if } x_N^c < w'. \tag{8}$$

Since the quadratic region of x_N^c is restricted to the region $|x_N^c| \leq \frac{\tau\rho_N(1-\rho_N)}{2}$, it follows that the quantity in (7) is less than or equal to $h_N - h_i$, and the quantity in (8) is greater than or equal to $b_i - b_N$. Hence, neither (7) nor (8) can equal zero unless h_i equals h_N for $x_N^c > w'$ or b_i equals b_N for $x_N^c < w'$. If the holding or backorder costs are equal, then the optimal x_N^c satisfies $|x_N^c| = \frac{\tau\rho_N(1-\rho_N)}{2}$, resulting in multiple optimal solutions along the line $x_i^c + x_N^c = w'$ with $|x_N^c| \geq \frac{\tau\rho_N(1-\rho_N)}{2}$ and $|x_i^c| \geq \frac{\tau\rho_i(1-\rho_i)}{2}$. Although many of these solutions lie in the region where products i and N are linear, one of the optimal solutions occurs when at least $N-1$ of the cost components are quadratic (or on the border between quadratic and linear). Hence, if there is a linear cost component at the optimal x^c, it will be $c_N(\tau, x_N^c, w)$. ∎

As a consequence, the optimal cycle center, or average amount of inventory per cycle, for product $i < N$ is restricted to the region $[-\tau\rho_i(1-\rho_i)/2, \tau\rho_i(1-\rho_i)/2]$, whereas product N's cycle center can be arbitrarily far from zero. Intuitively, this fact suggests that product N, which is the least cost product by our indexing convention, is the product that will hold the excess or deficit amounts of work when the total workload w fluctuates far from zero.

We now use Proposition 1 to find the optimal cycle center x^c. Without loss of generality, the workload constraint is used to eliminate x_N^c from the cost function, so that $x_N^c = w - \sum_{i=1}^{N-1} x_i^c$. To find the optimal center, we take the gradient of (5) and set it equal to zero:

$$\nabla_{x^c}[\sum_{i=1}^{N-1} c_i(\tau, x_i^c, w) + c_N(\tau, w - \sum_{i=1}^{N-1} x_i^c, w) + \frac{k}{\tau}] = 0. \tag{9}$$

At this point, we do not know whether $c_N(\tau, x_N^c, w)$ is linear or quadratic. Let $\bar{x}^c$ be the $(N-1)$-dimensional vector that solves (9) under the assumption that all N of the $c_i(\tau, x^c, w)$'s are quadratic in x_i^c. Taking the $(N-1)$-dimensional gradient, we find that $\bar{x}^c$ satisfies (w and τ multiply their vectors component-wise in the analysis below)

$$\frac{1}{\tau} A\bar{x}^c - \gamma_1 - \frac{w}{\tau}\gamma_2 = 0, \tag{10}$$

where

$$A = \begin{bmatrix} \ddots & & 0 \\ & \frac{b_i+h_i}{\rho_i(1-\rho_i)} & \\ 0 & & \ddots \end{bmatrix} + \frac{b_N+h_N}{\rho_N(1-\rho_N)} \begin{bmatrix} 1 & \cdots & 1 \\ \vdots & & \vdots \\ 1 & \cdots & 1 \end{bmatrix}, \tag{11}$$

$$\gamma_1 = \begin{bmatrix} \vdots \\ \frac{b_i-h_i}{2} - \frac{b_N-h_N}{2} \\ \vdots \end{bmatrix}, \tag{12}$$

$$\gamma_2 = \begin{bmatrix} \vdots \\ \frac{b_N+h_N}{\rho_N(1-\rho_N)} \\ \vdots \end{bmatrix}. \tag{13}$$

Thus,

$$\bar{x}^c = \tau A^{-1}\gamma_1 + wA^{-1}\gamma_2, \tag{14}$$

where the matrix elements of A^{-1} are

$$\alpha_{ij} = -\frac{\frac{\rho_i(1-\rho_i)}{b_i+h_i}\frac{\rho_j(1-\rho_j)}{b_j+h_j}}{\sum_{l=1}^{N}\frac{\rho_l(1-\rho_l)}{b_l+h_l}} \quad \text{for} \quad i \neq j, \quad \text{and} \tag{15}$$

$$\alpha_{ii} = \frac{\rho_i(1-\rho_i)}{b_i+h_i}\frac{\sum_{l=1}^{i-1}\frac{\rho_l(1-\rho_l)}{b_l+h_l} + \sum_{l=i+1}^{N}\frac{\rho_l(1-\rho_l)}{b_l+h_l}}{\sum_{l=1}^{N}\frac{\rho_l(1-\rho_l)}{b_l+h_l}}. \tag{16}$$

If $|w - \sum_{i=1}^{N-1}\bar{x}_i^c| \leq \frac{\tau\rho_N(1-\rho_N)}{2}$ then $c_N(\tau, x_N^c, w)$ is indeed quadratic and $\bar{x}_i^c$ determines the optimal center: $x_i^c = \bar{x}_i^c$ for $i < N$ and $x_N^c = w - \sum_{i=1}^{N-1}\bar{x}_i^c$. If $|w - \sum_{i=1}^{N-1}\bar{x}_i^c| > \frac{\tau\rho_N(1-\rho_N)}{2}$, we must solve the multivariate gradient equation with the linear form of $c_N(\tau, x_N^c, w)$. With this substitution, equation (9) decomposes into univariate expressions of the form

$$\frac{h_i - b_i}{2} + \frac{b_i + h_i}{\tau\rho_i(1-\rho_i)}x_i^c - h_N = 0 \quad \text{if } w - \textstyle\sum_{i=1}^{N-1}\bar{x}_i^c > \frac{\tau\rho_N(1-\rho_N)}{2}, \tag{17}$$

$$\frac{h_i - b_i}{2} + \frac{b_i + h_i}{\tau\rho_i(1-\rho_i)}x_i^c + b_N = 0 \quad \text{if } w - \textstyle\sum_{i=1}^{N-1}\bar{x}_i^c < -\frac{\tau\rho_N(1-\rho_N)}{2}. \tag{18}$$

Putting the results from (14) and (17)-(18) together, we obtain a complete expression for

the optimal cycle center. For $i < N$ we have

$$x_i^c = \begin{cases} \frac{\tau\rho_i(1-\rho_i)}{b_i+h_i}\left[\frac{b_i-h_i}{2} + h_N\right] & \text{if } w - \sum_{i=1}^{N-1} \bar{x}_i^c > \frac{\tau\rho_N(1-\rho_N)}{2} \\ \tau\alpha_i \cdot \gamma_1 + w\alpha_i \cdot \gamma_2 & \text{if } |w - \sum_{i=1}^{N-1} \bar{x}_i^c| \leq \frac{\tau\rho_N(1-\rho_N)}{2} \\ \frac{\tau\rho_i(1-\rho_i)}{b_i+h_i}\left[\frac{b_i-h_i}{2} - b_N\right] & \text{if } w - \sum_{i=1}^{N-1} \bar{x}_i^c < -\frac{\tau\rho_N(1-\rho_N)}{2} \end{cases} . \tag{19}$$

where α_i is the ith row of A^{-1}. The last component, x_N^c, is equal to $w - \sum_{i=1}^{N-1} x_i^c$.

The Optimal Cycle Time. The optimal cycle time τ can be derived by simple calculus. Substituting the optimal cycle center x_i^c into (4) yields the average inventory cost for each product as a function of τ and w. For $i < N$ this cost is given by

$$c_i(\tau, w) = \begin{cases} (b_i + h_i)\frac{\tau\rho_i(1-\rho_i)}{8} + \frac{\tau\rho_i(1-\rho_i)}{2(b_i+h_i)}\left[\frac{b_i-h_i}{2} + h_N\right]\left[\frac{h_i-b_i}{2} + h_N\right] & \text{I} \\ (b_i + h_i)\frac{\tau\rho_i(1-\rho_i)}{8} + \frac{\tau(b_i+h_i)}{2\rho_i(1-\rho_i)}(\alpha_i \cdot \gamma_1)^2 & \\ \quad + \frac{b_i+h_i}{\rho_i(1-\rho_i)}(\alpha_i \cdot \gamma_1)(\alpha_i \cdot \gamma_2)w + \frac{b_i+h_i}{2\tau\rho_i(1-\rho_i)}w^2(\alpha_i \cdot \gamma_2)^2 & \text{II} \\ \quad + \frac{h_i-b_i}{2}(\tau\alpha_i \cdot \gamma_1 + w\alpha_i \cdot \gamma_2) & \\ (b_i + h_i)\frac{\tau\rho_i(1-\rho_i)}{8} + \frac{\tau\rho_i(1-\rho_i)}{2(b_i+h_i)}\left[\frac{b_i-h_i}{2} - b_N\right]\left[\frac{h_i-b_i}{2} - b_N\right] & \text{III} \end{cases} , \tag{20}$$

where case I is shorthand for the region $w - \sum_{i=1}^{N-1} \bar{x}_i^c > \frac{\tau\rho_N(1-\rho_N)}{2}$, case II is $|w - \sum_{i=1}^{N-1} \bar{x}_i^c| \leq \frac{\tau\rho_N(1-\rho_N)}{2}$, and case III is $w - \sum_{i=1}^{N-1} \bar{x}_i^c < -\frac{\tau\rho_N(1-\rho_N)}{2}$. Similarly, product N's inventory cost function is

$$c_N(\tau, w) = \begin{cases} h_N\left(w - \tau\sum_{i=1}^{N-1} \frac{\rho_i(1-\rho_i)}{b_i+h_i}\left[\frac{b_i-h_i}{2} + h_N\right]\right) & \text{I} \\ (b_N + h_N)\frac{\tau\rho_N(1-\rho_N)}{8} + \frac{\tau(b_N+h_N)}{2\rho_N(1-\rho_N)}\left(\sum_{i=1}^{N-1} \alpha_i \cdot \gamma_1\right)^2 & \\ \quad - \frac{b_N+h_N}{\rho_N(1-\rho_N)}\left(\sum_{i=1}^{N-1} \alpha_i \cdot \gamma_1\right)w\left(1 - \sum_{i=1}^{N-1} \alpha_i \cdot \gamma_2\right) & \text{II} \\ \quad + \frac{b_N+h_N}{2\tau\rho_N(1-\rho_N)}w^2\left(1 - \sum_{i=1}^{N-1} \alpha_i \cdot \gamma_2\right)^2 & \\ \quad + \frac{h_N-b_N}{2}\left(w\left(1 - \sum_{i=1}^{N-1} \alpha_i \cdot \gamma_2\right) - \tau\sum_{i=1}^{N-1} \alpha_i \cdot \gamma_1\right) & \\ -b_N\left(w - \tau\sum_{i=1}^{N-1} \frac{\rho_i(1-\rho_i)}{b_i+h_i}\left[\frac{b_i-h_i}{2} - b_N\right]\right) & \text{III} \end{cases} . \tag{21}$$

The optimal value of τ is determined by taking the derivative of the total cost function

$\sum_{i=1}^{N} c_i(\tau, w) + k/\tau$ and solving against zero. If we define the constants

$$\xi_1 = \sum_{i=1}^{N-1} \frac{(b_i + h_i)\rho_i(1-\rho_i)}{8} - \frac{\rho_i(1-\rho_i)}{2(b_i+h_i)} \left(\frac{b_i - h_i}{2} + h_N \right)^2, \tag{22}$$

$$\xi_2 = \sum_{i=1}^{N} \frac{(b_i + h_i)\rho_i(1-\rho_i)}{8} + \sum_{i=1}^{N-1} \left[\frac{b_i + h_i}{2\rho_i(1-\rho_i)} (\alpha_i \cdot \gamma_1)^2 + \frac{h_i - b_i}{2} \alpha_i \cdot \gamma_1 \right] \tag{23}$$

$$+ \frac{b_N + h_N}{2\rho_N(1-\rho_N)} \gamma_4^2 - \frac{h_N - b_N}{2} \gamma_4, \tag{24}$$

$$\xi_3 = \sum_{i=1}^{N-1} \frac{(b_i + h_i)\rho_i(1-\rho_i)}{8} - \frac{\rho_i(1-\rho_i)}{2(b_i+h_i)} \left(\frac{b_i - h_i}{2} - b_N \right)^2, \tag{25}$$

$$\xi_4 = \sum_{i=1}^{N-1} \frac{b_i + h_i}{2\rho_i(1-\rho_i)} (\alpha_i \cdot \gamma_2)^2 + \frac{b_N + h_N}{2\rho_N(1-\rho_N)} \gamma_3^2, \tag{26}$$

where $\gamma_3 = 1 - \sum_{i=1}^{N-1} \alpha_i \cdot \gamma_2$ and $\gamma_4 = \sum_{i=1}^{N-1} \alpha_i \cdot \gamma_1$, then the optimal cycle length is given by

$$\tau = \begin{cases} \sqrt{\frac{k}{\xi_1}} & \text{I} \\ \sqrt{\frac{\xi_4 w^2 + k}{\xi_2}} & \text{II} \\ \sqrt{\frac{k}{\xi_3}} & \text{III} \end{cases}. \tag{27}$$

Considerable insight can be gleaned from (27). However, a discussion of the insights from our analysis is deferred until §1.8.

The Optimal Idling Threshold. With an expression for τ in hand, we can refine equation (3), which determines the idling threshold w_0. By using (27) to equate the cycle lengths at the borders of region II and its two adjacent regions, the three regions can be expressed in terms of the total workload level. Region I is characterized by workload levels $w > w_1$, region II by $w_2 \leq w \leq w_1$ and region III by $w < w_2$, where the workload level cutoffs are

$$w_1 = \sqrt{\frac{\xi_2 - \xi_1}{\xi_4 \xi_1} k}, \tag{28}$$

$$w_2 = -\sqrt{\frac{\xi_2 - \xi_3}{\xi_4 \xi_3} k}. \tag{29}$$

By (3) and (5), the total average cost is given by

$$\int_{-\infty}^{w_0} \left(\sum_{i=1}^{N} c_i(\tau, x_i^c, w) + \frac{k}{\tau} \right) \frac{2c}{\sigma^2} e^{-\frac{2c}{\sigma^2}(w_0 - w)} dw. \tag{30}$$

Having solved for τ in terms of the system parameters and w, we can substitute (27) into the inventory cost (20)-(21) and into the setup cost term k/τ in (30) to express the total average cost solely in terms of the total workload w; we will not explicitly write out this expression because it adds little to our understanding of the problem.

Depending on the idling threshold w_0, the integral can be broken into three or fewer terms based on the split of the total workload into three regions as defined in (28)-(29). Because of the irregular form of τ for $w \in [w_2, w_1]$, the integral over region II has no closed form solution. Therefore, we resort to numerical methods to determine the optimal value of w_0.

Our derivation of the optimal dynamic cyclic policy in heavy traffic is now complete. Recall that when the problem was initially defined, we assumed that product N had both the smallest holding and backorder cost indices. The analysis would remain unchanged, however, without this restriction. One only needs to introduce new notation (for example, $N-1$) to designate the product with the lowest backorder cost index, and use product $N-1$ in place of product N when the total workload level is less than zero.

1.7. The Proposed Policy. The final step in our analysis is to employ the optimal heavy traffic policy derived in §1.6 to develop a proposed policy for the original SELSP. This is done in two stages: We reverse the heavy traffic scalings to express the solution in terms of the original problem parameters, and then interpret the resulting solution.

If we replace the normalized quantities w, k and τ by $\tilde{w}/\sqrt{n}$, K/n and $\tilde{\tau}/\sqrt{n}$, respectively (τ undergoes this normalization because time is compressed by $\sqrt{n}$ in the fluid

model), then the optimal cycle length formula in (27) becomes

$$\tilde{\tau} = \begin{cases} \sqrt{\frac{K}{\xi_1}} & \text{I} \\ \sqrt{\frac{\xi_4 W^2 + K}{\xi_2}} & \text{II} \\ \sqrt{\frac{K}{\xi_3}} & \text{III} \end{cases} . \tag{31}$$

Motivated by (14), we define $\tilde{\tilde{x}}^c = \tilde{\tau} A^{-1}\gamma_1 + \tilde{w} A^{-1}\gamma_2$ for fixed total unnormalized workload $\tilde{W}(t) = \tilde{w}$. Then the unscaled cycle center $\tilde{x}^c$ is, for $i < N$,

$$\tilde{x}_i^c = \begin{cases} \frac{\tilde{\tau}\rho_i(1-\rho_i)}{b_i+h_i}\left[\frac{b_i-h_i}{2} + h_N\right] & \text{if } \tilde{w} - \sum_{i=1}^{N-1}\tilde{\tilde{x}}_i^c > \frac{\tilde{\tau}\rho_N(1-\rho_N)}{2} \\ \tilde{\tau}\alpha_i \cdot \gamma_1 + \tilde{w}\alpha_i \cdot \gamma_2 & \text{if } |\tilde{w} - \sum_{i=1}^{N-1}\tilde{\tilde{x}}_i^c| \leq \frac{\tilde{\tau}\rho_N(1-\rho_N)}{2} \\ \frac{\tilde{\tau}\rho_i(1-\rho_i)}{b_i+h_i}\left[\frac{b_i-h_i}{2} - b_N\right] & \text{if } \tilde{w} - \sum_{i=1}^{N-1}\tilde{\tilde{x}}_i^c < -\frac{\tilde{\tau}\rho_N(1-\rho_N)}{2} \end{cases} , \tag{32}$$

and $\tilde{x}_N^c$ is equal to $\tilde{w} - \sum_{i=1}^{N-1}\tilde{x}_i^c$. Hence, the unnormalized average inventory cost, $\tilde{c}_i(\tilde{\tau}, \tilde{x}^c, \tilde{w})$, is equal to $c_i(\tilde{\tau}, \tilde{x}^c, \tilde{w})$. Upon reversal of scaling, equations (28)-(29) become

$$\tilde{w}_1 = \sqrt{\frac{\xi_2 - \xi_1}{\xi_4\xi_1}K}, \tag{33}$$

$$\tilde{w}_2 = -\sqrt{\frac{\xi_2 - \xi_3}{\xi_4\xi_3}K}. \tag{34}$$

Therefore, if w_0 minimizes equation (30), then $\tilde{w}_0 = \sqrt{n}w_0$ will minimize the unnormalized long run average cost expression

$$\int_{-\infty}^{\tilde{w}_0}\left(\sum_{i=1}^{N}\tilde{c}_i(\tilde{\tau}, \tilde{x}_i^c, \tilde{w}) + \frac{K}{\tilde{\tau}}\right)\frac{2(1-\rho)}{\sigma^2}e^{-\frac{2(1-\rho)}{\sigma^2}(\tilde{w}_0 - \tilde{w})}d\tilde{w}. \tag{35}$$

Notice that the heavy traffic parameter n does not appear in (31)-(35). To compute $\tilde{w}_0$ in §3, we use Maple V to numerically solve the first-order optimality conditions associated with (35).

Our proposed dynamic cyclic policy for the SELSP must be expressed in terms of the original $(N+1)$-dimensional system state, which is given by the current inventory levels $\tilde{I}_1(t), \ldots, \tilde{I}_N(t)$ and the server location. There are many ways in which the unnormalized

policy in (31)-(32) and (35) can be interpreted for purposes of implementation. Perhaps the most natural way to express a dynamic lot-sizing policy is to specify a state-dependent maximum inventory (or "produce-up-to") level for the product currently being produced. By Figure 1, when the unnormalized total workload level $\tilde{W}(t)$ equals $\tilde{w}$, the maximum workload level for product i is $\tilde{x}_i^c + \rho_i(1-\rho_i)\tilde{\tau}/2$; of course, this level is achieved at the instant within a cycle when the production of product i is finished. Making use of the heavy traffic identity $I_i = \mu_i W_i$, let us define the three unnormalized workload regions in terms of the inventory process and the thresholds in (33)-(34): Region $\tilde{\mathrm{I}}$ is $\sum_{i=1}^N \mu_i^{-1}\tilde{I}_i(t) > \tilde{w}_1$, region $\tilde{\mathrm{II}}$ is $\sum_{i=1}^N \mu_i^{-1}\tilde{I}_i(t) \in [\tilde{w}_2, \tilde{w}_1]$ and region $\tilde{\mathrm{III}}$ is $\sum_{i=1}^N \mu_i^{-1}\tilde{I}_i(t) < \tilde{w}_2$. Then our proposed policy can be described as follows: *If $\sum_{i=1}^N \mu_i^{-1}\tilde{I}_i(t) > 0$ then let N refer to the product with the smallest holding cost index h_i; otherwise, let N denote the product with the smallest value of b_i. The server should idle if $\sum_{i=1}^N \mu_i^{-1}\tilde{I}_i(t) > \tilde{w}_0$; otherwise, if set up for product $i < N$ then produce this product as long as*

$$\mu_i^{-1}\tilde{I}_i(t) < \begin{cases} \sqrt{\rho_i(1-\rho_i)\frac{K}{\xi_1}\frac{b_i+h_N}{b_i+h_i}} & \tilde{\mathrm{I}} \\ \sqrt{\frac{\xi_4(\sum_{i=1}^N \mu_i^{-1}\tilde{I}_i(t))^2+K}{\xi_2}}(\alpha_i\cdot\gamma_1 + \frac{\rho_i(1-\rho_i)}{2}) + (\alpha_i\cdot\gamma_2)\sum_{i=1}^N \mu_i^{-1}\tilde{I}_i(t) & \tilde{\mathrm{II}} \\ \sqrt{\rho_i(1-\rho_i)\frac{K}{\xi_3}\frac{b_i-b_N}{b_i+h_i}} & \tilde{\mathrm{III}} \end{cases} . \tag{36}$$

Once $\mu_i^{-1}\tilde{I}_i(t)$ reaches or exceeds this level, switch to the next product. If set up for product N, then produce this product while

$$\mu_N^{-1}\tilde{I}_N(t) < \begin{cases} \sum_{i=1}^N \mu_i^{-1}\tilde{I}_i(t) + \sqrt{\frac{K}{\xi_1}}(\frac{\rho_N(1-\rho_N)}{2} - \sum_{i=1}^{N-1}\frac{\rho_i(1-\rho_i)}{b_i+h_i}\left[\frac{b_i-h_i}{2}+h_N\right]) & \tilde{\mathrm{I}} \\ \sqrt{\frac{\xi_4(\sum_{i=1}^N \mu_i^{-1}\tilde{I}_i(t))^2+K}{\xi_2}}(\frac{\rho_n(1-\rho_N)}{2} - \sum_{i=1}^{N-1}\alpha_i\cdot\gamma_1) & \tilde{\mathrm{II}} \\ +(\sum_{i=1}^N \mu_i^{-1}\tilde{I}_i(t))(1-\sum_{i=1}^{N-1}\alpha_i\cdot\gamma_2) & \\ \sum_{i=1}^N \mu_i^{-1}\tilde{I}_i(t) + \sqrt{\frac{K}{\xi_3}}(\frac{\rho_N(1-\rho_N)}{2} - \sum_{i=1}^{N-1}\frac{\rho_i(1-\rho_i)}{b_i+h_i}\left[\frac{b_i-h_i}{2}-b_N\right]) & \tilde{\mathrm{III}} \end{cases} , \tag{37}$$

and then switch to the next product when $\mu_N^{-1}\tilde{I}_N(t)$ reaches or exceeds this level.

1.8. Insights. Our analysis reveals numerous insights into the behavior of the optimal policy in heavy traffic. Before reading this subsection, readers may want to

digest the graphs marked "proposed" in Figures 3 and 4 in §3.2, which depict the proposed policies in both the symmetric (each product has the same parameters) and asymmetric two-product settings.

Three Workload Regions. An essential feature of the heavy traffic policy is its characterization via three workload regions, as described in (28)-(29). There is sufficient workload in region I, significant backorders in region III, and region II represents the intermediate case where the total workload is in an interval containing zero.

State-Dependent Lot Sizes. Because the time spent producing product i within a cycle is $\rho_i\tau$, the optimal cycle length τ determines the optimal lot sizes in heavy traffic (and determines the optimal *expected* lot sizes for the SELSP). We can observe from (27) that the optimal lot sizes are state-dependent when the total workload is in region II. In contrast, the lot sizes are constant in regions I and III; in these regions, surplus or deficit inventory is unavoidable, and the trade-off between lot sizes and setup costs stabilize, thereby generating constant lot sizes. This observation and (19) imply that the cycle center x^c remains constant in regions I and III, and gradually shifts between these two points in the intermediate area of region II. It is worth pointing out that in nearly all of the deterministic ELSP literature (Dobson 1987 is a notable exception), the analysis is restricted to policies with constant lot sizes.

Relationship to the EOQ Model. As in the economic order quantity (EOQ) model, the lot size is proportional to the square root of the setup cost in regions I and III. In region II, the setup cost again appears in the numerator of the square root term.

Inventory is Focused in the Least Cost Products. In region I, excess inventory is built up in the product with the smallest h_i, which is a product that is inexpensive to hold (small $\bar{h}_i$) and lengthy to process (small μ_i). Similarly, in Region III, excess negative inventory (i.e., backorders) is held in the product with the smallest backorder cost index b_i; this product is inexpensive to backorder and has a long expected processing time. In both regions, inventory is held in the least cost product so as to reduce the absolute value of the inventory of the higher (holding in region I and backorder in region III) cost products. In this regard, the dynamic lot-sizing policy derived here is similar to the heavy traffic policy derived for the corresponding problem without setups in Wein (1992). In

that paper, instantaneous switching causes the inventories of all the higher cost products to vanish in the heavy traffic normalization. When setup costs are introduced, breadth is added to the normalized cycle length and, for a fixed total workload, a "corridor" of possible inventory states replaces the least cost axes. In fact, if we consider the special case $K = 0$, then region I (region III) corresponds to $w > 0 (w < 0)$; in both regions, $\tau = 0$ and $x_N^c = w$, and the solution reverts to that of Wein.

Lot Sizes Grow with Absolute Value of Total Workload. By (27), we see that the optimal lot size is smallest when the total workload equals zero, and grows with the absolute value of the workload. When the total workload is near zero, costly backorders can be avoided by switching frequently between products. In contrast, when the absolute value of the workload is large, it is possible to employ large lot sizes without adversely affecting the inventory costs (because inventory tends to be held in the minimum cost product in regions I and III); in this case, it is advantageous to avoid setup costs and produce products in large batches.

Inventory Levels at Switching Epochs. In heavy traffic, the maximum normalized workload for product i under our proposed policy is $x_i^c + \rho_i(1-\rho_i)\tau/2$. It follows that for $i < N$, product i inventory is negative (i.e., backordered) when the server switches into product i and is positive when the server switches out of product i. For product N the sign of the inventory level during the switching epochs depends on the region: In region II product N inventory is backordered when the server switches into product N and is positive when the server switches out of product N. In contrast, product N inventory is always positive in region I and is always negative in region III.

Cost-Symmetric Case. Our results simplify considerably under the *cost-symmetric* case, where $h_i = h$ and $b_i = b$ for all $i = 1, \ldots, N$. This case will arise, for example, if all products are relatively indistinguishable, except for their color. Then each product would be expected to have the same service rate and inventory cost rates, but different demand rates, since some colors may be more popular than others. Then $\xi_1 = \xi_3 = 0$ and, by (28)-(29), the workload always resides in region II. In fact, one can show that the workload resides in only one region (region II) if and only if the costs are symmetric. If we define the constant $\hat{\rho} = \sum_{i=1}^N \rho_i(1-\rho_i)$, then when the server is busy and is set up for

product i, this product is produced as long as

$$\mu_i^{-1}\tilde{I}_i(t) < \rho_i(1-\rho_i)\left[\sqrt{\frac{(\sum_{i=1}^N \mu_i^{-1}\tilde{I}_i(t))^2}{\hat{\rho}^2} + \frac{2K}{\hat{\rho}(b+h)}} + \frac{\sum_{i=1}^N \mu_i^{-1}\tilde{I}_i(t))}{\hat{\rho}}\right]. \quad (38)$$

It is evident from this expression that the cost structure effects the lot sizes only through the ratio $K/(b+h)$; not surprisingly, the lot size is an increasing function of this quantity.

2. THE SETUP TIME PROBLEM

2.1. Problem Description. In the setup time problem, a random setup time (rather than a setup cost) is incurred when the server switches from one product to another. In all other respects, the setup cost problem and the setup time problem are identical, and all relevant notation from §1 will be retained. As in the setup cost problem, we shall only consider dynamic cyclic policies. Let s denote the mean setup time per cycle; Coffman, Puhalskii and Reiman (1995b) show that the heavy traffic performance of this system depends upon the setup time distributions only via this quantity, and other key parameters are unaffected by a permutation in the cycle order. Thus in heavy traffic the desired permutation is the travelling salesman tour, where the intercity distances are given by the mean setup times. Under general traffic conditions, Poisson arrivals and deterministic service times, Fuhrmann (1992) and Cooper, Niu, and Srinivasan (1992) show that the mean waiting time depends on setup times only through the total setup time over the cycle. Also, numerical calculations in Federgruen and Katalan (1993) show that the performance of a cyclic policy is quite insensitive to the chosen permutation. The server has three scheduling options at each point in time: Produce a unit of the product that is currently set up, initiate a setup for the next product in the cycle or sit idle. The control problem is to find a nonanticipating scheduling policy to minimize

$$\limsup_{T\to\infty} \frac{1}{T} E\Big[\int_0^T \sum_{i=1}^N \Big(\bar{h}_i\tilde{I}_i^+(t) + \bar{b}_i\tilde{I}_i^-(t)\Big)\,dt\Big]. \quad (39)$$

Once again, we avoid unnecessary notation by omitting the equations that express the inventory vector $\tilde{I}$ in terms of an arbitrary cyclic policy.

2.2. The Diffusion Control Problem. The heavy traffic normalizations are the same as in §2, and the setup times are not rescaled in heavy traffic; that is, they are $O(1)$. The form of the proposed policy is also the same as before. The server idles if $W(t) \geq w_0$ and works otherwise, where w_0 is an unknown normalized threshold parameter. When the server is busy and the total workload equals w, the cycle time is $\tau(w)$ and the vector of average inventory workloads is $x^c(w)$.

Since the setup times are $O(1)$, they occur instantaneously in the heavy traffic limit, and Coffman, Puhalskii and Reiman (1995b) show that the time scale decomposition described in §1.3 also carries over to the setup time problem. More specifically, when the total normalized workload $W(t)$ equals w, the N-dimensional fluid process $(\bar{W}_1, \ldots, \bar{W}_N)$ is identical to the N-dimensional fluid process in the setup cost problem; hence, the optimal placement of cycle center x^c is given by equation (19) and the inventory cost rate $c_i(x^c, \tau, w)$ is defined in equations (20)-(21).

We now characterize the normalized total workload process W for the setup time problem under the class of dynamic cyclic policies. Our characterization of the total workload process follows from existing heavy traffic limit theorems in Coffman, Puhalskii and Reiman (1995b) after making the innocuous assumption that the time scale decomposition holds for all multiproduct cyclic policies, not just for two-product cyclic base stock policies. For make-to-order systems, readers are referred to Coffman, Puhalskii and Reiman (1995b) for a derivation of this characterization in the two-product uncontrollable case, and to Reiman and Wein for a discussion of the controllable case; here we only state the approximation. The normalized total workload process W is assumed to be a diffusion process. Its variance σ^2 is the same as in the setup cost problem, and is defined in (2). The drift of W is no longer $\sqrt{n}(1-\rho)$, but $\sqrt{n}(f(w)-\rho)$, where $f(w)$ is the fraction of time during a cycle that the server serves customers, as opposed to incurring setups, when the total workload equals w. When the normalized workload equals w, this fraction is given by

$$f(w) = \frac{\sqrt{n}\tau(w)}{\sqrt{n}\tau(w) + s}, \tag{40}$$

where $\tau(w)$ is the cycle length, which is both deterministic and controllable. Hence,

$\sqrt{n}(f(w) - \rho) \to c - s/\tau(w)$ as $n \to \infty$, and we assume that W has the state-dependent drift

$$\mu(w) = c - \frac{s}{\tau(w)}. \tag{41}$$

Notice that this drift approaches the drift in the setup cost problem as the setup times vanish or the cycle length becomes arbitrarily large.

Define the cost function $c(\tau, w) = \sum_{i=1}^{N} c_i(\tau, w)$, where the individual cost components are defined in (20)-(21). The approximating diffusion control problem is to choose the state-dependent cycle length $\tau(w) \geq 0$ and the threshold w_0 to minimize

$$\limsup_{T\to\infty} \frac{1}{T} E\Big[\int_0^T c(\tau(W(t)), W(t))dt\Big], \tag{42}$$

where W is a $(\mu(w), \sigma^2)$ reflected diffusion process on $(-\infty, w_0]$. Hence, the controllable cycle length $\tau(w)$ affects both the drift $\mu(w)$ and the cost $c(\tau, w)$ in a nonlinear fashion.

2.3. The Optimality Conditions. Problem (42) involves a drift control $\tau(w)$ and a singular control via the reflecting barrier w_0. Although the drift is unbounded as $\tau(w) \to 0$, we proceed as if standard arguments apply (Mandl 1968) and state the Hamilton-Jacobi-Bellman optimality conditions:

$$\min_{\tau(w)\geq 0} \left\{ c(\tau(w), w) - g + \left(c - \frac{s}{\tau(w)}\right) V'(w) + \frac{\sigma^2}{2} V''(w) \right\} = 0 \quad \text{for } w \leq w_0 \tag{43}$$

and

$$V'(w) = 0 \quad \text{for } w \geq w_0, \tag{44}$$

where g is the gain and $V(x)$ is the potential (relative value) function. If a solution $(g, V(w), \tau(w), w_0)$ can be found to (43)-(44), then $(\tau(w), w_0)$ is the optimal solution to (42), g is the optimal long run average cost (independent of initial state), and $V(w)$ is the cost incurred under the optimal policy when the initial state is w minus the cost incurred when the initial state is some reference state, which we take to be w_0. We assume that $V \in C^2$, and define $p(w) = V'(w)$; this assumption, which is known as the heuristic principle of smooth fit (Beneš, Shepp and Witsenhausen 1980), is often imposed when solving diffusion control problems.

Using the workload cutoff levels w_1 and w_2 to distinguish among regions I, II and III (w_1 and w_2 are unknown at this point and are not given by (28)-(29)), we substitute the three forms of cost function $c(\tau, w)$ into (43) to obtain

$$\min_{\tau(w)\geq 0}\left\{\xi_1\tau(w)+h_N w-g+\left(c-\frac{s}{\tau(w)}\right)p(w)+\frac{\sigma^2}{2}p'(w)\right\}=0 \quad \text{for } w_1\leq w\leq w_0, \tag{45}$$

$$\min_{\tau(w)\geq 0}\left\{\xi_2\tau(w)+\xi_5 w+\xi_4\frac{w^2}{\tau(w)}-g+\left(c-\frac{s}{\tau(w)}\right)p(w)+\frac{\sigma^2}{2}p'(w)\right\}=0 \quad \text{for } w_2\leq w\leq w_1. \tag{46}$$

$$\min_{\tau(w)\geq 0}\left\{\xi_3\tau(w)-b_N w-g+\left(c-\frac{s}{\tau(w)}\right)p(w)+\frac{\sigma^2}{2}p'(w)\right\}=0 \quad \text{for } w\leq w_2, \tag{47}$$

where the new constant ξ_5 is

$$\xi_5=\sum_{i=1}^{N-1}\left[\frac{b_i+h_i}{\rho_i(1-\rho_i)}(\alpha_i\cdot\gamma_1)(\alpha_i\cdot\gamma_2)+\frac{h_i-b_i}{2}\alpha_i\cdot\gamma_2\right]-\frac{b_N+h_N}{\rho_N(1-\rho_N)}\gamma_4\gamma_3+\frac{h_N-b_N}{2}\gamma_3. \tag{48}$$

Solving the first-order optimality conditions for $\tau(w)$ yields

$$\tau^*(w)=\begin{cases}\sqrt{-\frac{sp(w)}{\xi_1}} & \text{for } w_1\leq w\\ \sqrt{\frac{\xi_4}{\xi_2}w^2-\frac{sp(w)}{\xi_2}} & \text{for } w_2\leq w\leq w_1\\ \sqrt{-\frac{sp(w)}{\xi_3}} & \text{for } w\leq w_2\end{cases}. \tag{49}$$

Substituting $\tau^*(w)$ into (45)-(47), we obtain the nonlinear ordinary differential equations (ODE's)

$$2\sqrt{-\xi_1 sp(w)}+h_N w-g+cp(w)+\frac{\sigma^2}{2}p'(w)=0 \qquad \text{for } w_1\leq w\leq w_0. \tag{50}$$

$$2\sqrt{\xi_2\xi_4 w^2-\xi_2 sp(w)}+\xi_5 w+cp(w)+\frac{\sigma^2}{2}p'(w)-g=0 \qquad \text{for } w_2\leq w\leq w_1, \tag{51}$$

$$2\sqrt{-\xi_3 sp(w)}-b_N w-g+cp(w)+\frac{\sigma^2}{2}p'(w)=0 \qquad \text{for } w\leq w_2. \tag{52}$$

2.4 Structural Properties. Before pursuing a solution to the diffusion control problem, we derive several useful structural properties.

Property 1. *If $w_2 = -\infty$ then*

$$p(w) = \frac{2\sqrt{\xi_2\xi_4} - \xi_5}{c} w + o(w) \quad \text{as} \quad w \to -\infty; \tag{53}$$

if $w_2 \neq -\infty$ then

$$p(w) = -\frac{b_N}{c} w + o(w) \quad \text{as} \quad w \to -\infty. \tag{54}$$

The derivation of this property, which assumes an asymptotic monotonicity property, is nearly identical to that in the Appendix of Reiman and Wein (1994); see Markowitz (1995) for details.

Property 2. *The idling threshold w_0 is greater than or equal to w_1, and equality holds if and only if $h_i = h_j$ for all i and j. At the idling threshold, the cycle length $\tau^*(w_0) = 0$ if $w_0 > w_1$ and $\tau^*(w_0) = \sqrt{\xi_4/\xi_2}w_0$ if $w_0 = w_1$.*

Proof: The boundary condition $p(w_0) = 0$ and equation (49) imply that $\tau^*(w_0) = 0$ if $w_0 > w_1$ and $\tau^*(w_0) = \sqrt{\xi_4/\xi_2}w_0$ if $w_0 \leq w_1$. In order for $w_0 \leq w_1$, $\tau^*(w_0)$ must satisfy the cycle placement conditions originally specified in the conditioning of equation (19). We can rewrite the centering condition as

$$|w_0 - \sum_{i=1}^{N-1} \tau^*(w_0)\alpha_i \cdot \gamma_1 + w_0\alpha_i \cdot \gamma_2| \leq \frac{\tau^*(w_0)\rho_N(1-\rho_N)}{2}. \tag{55}$$

Rearranging terms, this inequality is

$$\gamma_3 \leq \sqrt{\frac{\xi_4}{\xi_2}}(\gamma_4 + \frac{\rho_N(1-\rho_N)}{2}). \tag{56}$$

Algebraic manipulations (see Markowitz) show that this inequality is false if inventory holding costs are not identical and holds at equality only if $h_i = h_j$ for all i and j. In the cost-symmetric case, the placement conditions in equation (55) are satisfied at equality by the idling threshold w_0, and hence w_0 is the boundary between regions (I) and (II); that is, $w_0 = w_1$. ∎

Property 3. $w_2 = -\infty$ *if and only if $b_i = b_j$ for all i and j.*

Derivation: The constant ξ_3 equals 0 in the cost-symmetric case. This fact and (49) imply that $\tau^*(w)$ is unbounded for $w < w_2$, which would violate the placement conditions. Hence, $w_2 = -\infty$ in the cost-symmetric case. If $w_2 = -\infty$ then $p(w)/w \to \frac{2\sqrt{\xi_2\xi_4}-\xi_5}{c}$ as $w \to -\infty$ by Property 1, and $\tau^*(w)/w \to -\sqrt{\frac{\xi_4}{\xi_2}}$ as $w \to -\infty$ by (49). To remain in region (II), the placement condition

$$|w - \sum_{i=1}^{N-1} \tau^*(w)\alpha_i \cdot \gamma_1 + w\alpha_i \cdot \gamma_2| \leq \frac{\tau^*(w)\rho_N(1-\rho_N)}{2} \tag{57}$$

must be satisfied. For w approaching $-\infty$, we can substitute the limit of $\tau^*(w)$ and by algebraic manipulation reduce equation (57) to

$$\gamma_3 \leq \sqrt{\frac{\xi_4}{\xi_2}}(\frac{\rho_N(1-\rho_N)}{2} - \gamma_4). \tag{58}$$

As in Property 2, this inequality is false if backorder costs are not identical and holds at equality if $b_i = b_j$ for all i and j (see Markowitz for details). ∎

The nonlinear ODE's in (50)-(52) do not appear to admit a closed form solution. As in Reiman and Wein, we pursue both an approximate analytical solution and a numerical solution. The approximate analytical solution aims to compute a function $p(x)$ that nearly satisfies the optimality conditions (44) and (50)-(52). The basic idea behind our method is to use Taylor series expansions to linearize the square root terms in (50)-(52), and then to use Properties 1 through 3 to paste together the solutions of the resulting ODE's so as to ensure that the principle of smooth fit holds. However, the scheduling policy arising from this analysis did not perform consistently well over the test cases considered in §3. Hence, this approximate analysis and the corresponding numerical results are not given here, and the interested reader is referred to Markowitz for details.

The next subsection contains insights from our structural results, and an algorithmic procedure for solving the diffusion control problem is provided in §2.6.

2.5. Insights. Because a closed form solution is not obtained for the setup time problem, it is more difficult to develop insights into the behavior of the optimal solution. Nevertheless, several noteworthy comparisons can be made.

Cost-Symmetry vs. Cost-Asymmetry. The solution for these two cases are surprisingly different; see the graphs entitled "proposed" in Figures 5 and 6 in §3.3. In the cost-symmetric case the workload stays in region II, and the cycle length at the idling threshold, $\tau(w_0)$, is proportional to w_0. In the cost-asymmetric case, the policy is characterized by three regions, and the lot size approaches zero as the workload approaches the idling threshold; these results follow from Properties 2 and 3. Evidently, near the idling threshold, small lot sizes are used in the asymmetric case to reduce inventory costs, whereas this option is not available in the cost-symmetric case. Finally, by Properties 1 and 3 and equation (49), as the current total workload w tends to $-\infty$, the cycle length grows with $\sqrt{-w}$ in the asymmetric case and with $-w$ in the symmetric case. Hence, when backorders are large, there is less opportunity to reduce inventory costs in the cost-symmetric case, and larger lot sizes prevail in order to reduce the amount of time devoted to setups.

Setup Costs vs. Setup Times. It is interesting to note both the similarities and differences between the setup cost and setup time problems: The behavior of the proposed policies for both problems can be distinctly broken down into three workload regions (one region, respectively) when costs are asymmetric (symmetric, respectively). For both problems, lot sizes are state-dependent and inventory is focused in the least cost products; moreover, the description of the inventory levels at the switching epochs in §1.8 carries over to the setup time problem. The proposed policies for the two problems are qualitatively similar in the region around $w = 0$ (i.e., region II), but have different characteristics in the two extreme regions (regions I and III). In the asymmetric setup cost problem, the cycle time τ remains constant throughout these two regions. In the asymmetric setup time problem, the cycle time contracts to zero as w approaches the idling threshold and grows as $\sqrt{-w}$ when w tends to $-\infty$.

As noted in Reiman and Wein, the two problems lead to qualitatively different solutions because queueing effects cause setup times to consume available capacity in a highly nonlinear manner. Therefore, *the effective cost of a setup time is workload-dependent in the setup time problem*: There is no direct penalty for a set up, only an increased probability that the total workload will fall. As the total workload approaches $-\infty$, many items

are backordered and the effective cost of a setup is very high; thus, the scheduler attempts to use the capacity efficiently by running large lot sizes, so as to recover from the low workload level. In contrast, as the total workload approaches the idling threshold w_0, the effective cost of a setup time decreases, and the scheduler can afford to employ small lot sizes to reduce inventory costs. As a consequence of our analysis, it is clear that setup costs should not be used as a surrogate for setup times in the SELSP; unfortunately, this practice is quite common in the deterministic ELSP literature.

Setup Times vs. *No Setup Times.* Like the setup cost policy, the proposed setup time policy is a generalization of Wein. As setup times vanish, all of the inventory becomes stored only in product N. Using the symmetric cost results, the optimal idling threshold w_0 goes to $-\ln(\frac{h}{b+h})\sigma^2/(2c)$, which is the same as that derived by Wein.

2.6 An Algorithmic Solution. Since problem (42) cannot be solved analytically, we pursue a numerical solution. In particular, the *Markov chain approximation* technique developed by Kushner (1977) is employed. This method systematically discretizes both time and the state space, and approximates a diffusion control problem by a control problem for a finite state Markov chain. Weak convergence methods have been developed by Kushner and his colleagues to verify that the controlled Markov chain (and its corresponding optimal cost) approximates arbitrarily closely the controlled diffusion process (and its corresponding optimal cost); we refer readers to Kushner and Dupuis (1992) for an up-to-date account of this research area, and retain most of their notation for ease of reference. It is worth emphasizing that the *computational complexity of the algorithmic approach is independent of the number of products*; this important fact is due to the state space collapse inherent in the CPR result, which leaves us with a one-dimensional diffusion process, and our optimization of the cycle center in (19).

Before describing the method, we introduce a slight modification to the heavy traffic analysis to account for the fact that setup times do not vanish in the original problem. The cycle length $\tau(w)$ consists of the time devoted to processing and the time allocated to setups. In the fluid scaling, $s/\sqrt{n}$ units of time are spent setting up over the course of a cycle; although this quantity vanishes in the limit, we include it in our analysis as an intended refinement. More specifically, we replace $\tau(w)$ by $\tau(w) + s/\sqrt{n}$.

Let h denote the *finite difference interval*, which dictates how finely both the state space and time are discretized. One can consider a sequence of controlled Markov chains indexed by the interval h, and as the value of h becomes smaller the resulting discrete time, finite state Markov chain described below becomes a better approximation of the controlled diffusion process.

To numerically solve (42), we need to confine the one-dimensional diffusion process W to a bounded region. Since W resides on a halfline, the state space of the controlled Markov chain will be $\{-M, -M+h, ..., M-h, M\}$, where M is a positive integer multiple of h. For now let us fix the idling threshold w_0 (this parameter will be optimized later on) such that $w_0 \leq M$ and w_0 is an integer multiple of h. Hence, the Markov chain actually resides in $\{-M, -M+h, ..., w_0-h, w_0\}$. The approximating Markov chain has nonzero transition probabilities

$$P^h(w, w+h) = \frac{\sigma^2 + 2h\left(c - \frac{s}{\tau(w)+s/\sqrt{n}}\right)^+}{2\sigma^2 + 2h\left|c - \frac{s}{\tau(w)+s/\sqrt{n}}\right|} \tag{59}$$

and

$$P^h(w, w-h) = \frac{\sigma^2 + 2h\left(c - \frac{s}{\tau(w)+s/\sqrt{n}}\right)^-}{2\sigma^2 + 2h\left|c - \frac{s}{\tau(w)+s/\sqrt{n}}\right|} \tag{60}$$

on the interior of the state space, and the time intervals, or *interpolation intervals*, are of length

$$\Delta t^h = \frac{h^2}{\sigma^2 + h\left|c - \frac{s}{\tau(w)+s/\sqrt{n}}\right|} . \tag{61}$$

Two issues need to be addressed to obtain our approximating controlled Markov chain: (i) for an ergodic cost problem, the interpolation interval Δt^h must be independent of the state w and control $\tau(w)$ (see Kushner and Dupuis, page 209), and (ii) the behavior of the Markov chain at the boundary states $w = -N$ and $w = w_0$. To deal with the first issue, we define $Q^h = \sigma^2 + \max_{w,\tau(w)} h\left|c - \frac{s}{\tau(w)+s/\sqrt{n}}\right|$. Since the smallest nonzero value of $\tau(w)$ is h, we let $Q^h = \sigma^2 + |ch - s|$, and define the new nonzero interior transition

probabilities

$$\bar{P}^h(w, w+h) = \frac{\sigma^2 + 2h\left(c - \frac{s}{\tau(w)+s/\sqrt{n}}\right)^+}{2Q^h} , \tag{62}$$

$$\bar{P}^h(w, w-h) = \frac{\sigma^2 + 2h\left(c - \frac{s}{\tau(w)+s/\sqrt{n}}\right)^-}{2Q^h} \tag{63}$$

and

$$\bar{P}^h(w, w) = 1 - \frac{\left(\sigma^2 + h\left|c - \frac{s}{\tau(w)+s/\sqrt{n}}\right|\right)}{Q^h} , \tag{64}$$

and the new interpolation interval

$$\Delta t^h = \frac{h^2}{Q^h} . \tag{65}$$

Now we consider the boundary states. A reflecting boundary is employed at the idling threshold. However, the Markov chain approximation method assumes that $\Delta t^h = 0$ for a reflecting boundary state. Because the interpolation interval Δt^h takes on a value different than (65) at w_0, this boundary state must be eliminated. We define the transition probability (see page 212 of Kushner and Dupuis)

$$\tilde{P}^h(w_0 - h, w_0 - h) = 1 - \bar{P}^h(w_0 - h, w_0 - 2h) . \tag{66}$$

We also impose a reflecting boundary at state $-M$, and define the transition probability

$$\tilde{P}^h(-M + h, -M + h) = 1 - \bar{P}(-M + h, -M + 2h) . \tag{67}$$

Although the reflecting barrier at $-M$ is artificial in the sense that $P(-M, -M + h)$ would be positive if the boundary was chosen to be larger than M, the effect of this approximation should be negligible if the value of M is sufficiently large, and consequently visited sufficiently infrequently. In our implementation, the size of the Markov chain is chosen so that a further increase in M does not change the optimal solution $(\tau(w), w_0)$. In summary, our approximating Markov chain has state space $\{-M + h, -M + 2h, ..., w_0 - 2h, w_0 - h\}$, interpolation interval defined by (65), and nonzero transition probabilities

$\tilde{P}^h(w,y)$ defined by (66)–(67) and $\tilde{P}^h(w,y) = \bar{P}^h(w,y)$ otherwise, where $\bar{P}^h(w,y)$ are defined in equations (62)-(64).

The dynamic programming optimality equation for the controlled Markov chain is given by (see equation 5.3 on page 204 of Kushner and Dupuis)

$$V(w) = \begin{cases} \sum_y \tilde{P}^h(w,y)V(y) + (\xi_1(\tau(w) + s/\sqrt{n}) + h_N w - g)\Delta t^h & \text{for } w_1 \le w \le w_0 \\ \sum_y \tilde{P}^h(w,y)V(y) + (\xi_2(\tau(w) + s/\sqrt{n}) + \xi_5 w \\ \quad + \xi_4 \frac{w^2}{\tau(w)+s/\sqrt{n}} - g)\Delta t^h & \text{for } w_2 \le w \le w_1 \\ \sum_y \tilde{P}^h(w,y)V(y) + (\xi_3(\tau(w) + s/\sqrt{n}) - b_N w - g)\Delta t^h & \text{for } -N \le w \le w_2 \end{cases} . \tag{68}$$

and the Markov chain control problem can be solved using the following policy improvement algorithm. First, we choose the initial policy: $\tau(w) = w_0 - w$ for $w \le w_0$ and arbitrary w_0. In the evaluation step of the algorithm, a generic policy $(\tau(w), w_0)$ is evaluated (that is, $V(w)$ and g are found) recursively. Since the Markov chain is a birth-death process, the stationary probability distribution π_w for $w \in \{-M+h, ..., M-h\}$ is

$$\pi_w = \begin{cases} 0 & \text{for } w > w_0 - h \\ \pi_{w_0-h} \prod_{k=w+1}^{w_0-h} \frac{\tilde{P}^h(k,k-h)}{\tilde{P}^h(k-h,k)} & \text{for } w < w_0 - h \\ \left(1 + \sum_{l=-N+h}^{w_0-h} \prod_{k=l+h}^{w_0-h} \frac{\tilde{P}^h(k,k-h)}{\tilde{P}^h(k-h,k)}\right)^{-1} & \text{for } w = w_0 - h \end{cases} . \tag{69}$$

and the gain is

$$g = \sum_{w=-N+h}^{w_0-h} \pi_w c(\tau(w) + s/\sqrt{n}, w). \tag{70}$$

We set $V(M) = 0$, so that $V(w) = 0$ for $w \ge w_0$, by equation (44). For $w < w_0$, equation (68) implies that $V(w-h)$ can be calculated recursively by

$$V(w-h) = \frac{g - c(\tau(w), w) + (1 - \tilde{P}^h(w,w))V(w) - \tilde{P}^h(w,w+h)V(w+h)}{\tilde{P}^h(w,w-h)}. \tag{71}$$

In the policy improvement step we first solve for the cycle length $\tau(w)$ and then for the idling threshold w_0. The cycle length is determined by minimizing the right side of (68)

with respect to $\tau(w) \geq 0$. If the drift $c - s/(\tau^*(w) + s/\sqrt{n})$ is positive then

$$\tau^*(w) = \begin{cases} \sqrt{-\frac{s[V(w+h)-V(w)]}{\xi_1 h}} & \text{for } w_1 \leq w \leq w_0 \\ \sqrt{\frac{\xi_4}{\xi_2} w^2 - \frac{s[V(w+h)-V(w)]}{\xi_2 h}} & \text{for } w_2 \leq w \leq w_1 \\ \sqrt{-\frac{s[V(w+h)-V(w)]}{\xi_3 h}} & \text{for } -N \leq w \leq w_2 \end{cases} . \tag{72}$$

and if the drift is negative then

$$\tau^*(w) = \begin{cases} \sqrt{-\frac{s[V(w)-V(w-h)]}{\xi_1 h}} & \text{for } w_1 \leq w \leq w_0 \\ \sqrt{\frac{\xi_4}{\xi_2} w^2 - \frac{s[V(w)-V(w-h)]}{\xi_2 h}} & \text{for } w_2 \leq w \leq w_1 \\ \sqrt{-\frac{s[V(w)-V(w-h)]}{\xi_3 h}} & \text{for } -N \leq w \leq w_2 \end{cases} . \tag{73}$$

Notice that (72)–(73) converges to (49) as $h \to 0$, as expected. Keeping $\tau(w)$ constant, we determine an improved idling threshold w_0 by evaluating (via equation (70)) the gain g for all values of w_0, and searching for the gain-minimizing value.

The evaluation and improvement steps of the policy improvement algorithm are repeated until the new and old gains are sufficiently close in value and the idling threshold remains constant. In the implementation of this algorithm in §3, we chose $M = 1000$ and terminated the algorithm when the difference between the new and old gains was less than 0.05 and the idling thresholds changed less than 0.1; our choice for the finite difference interval h is discussed in the next paragraph. The output of our algorithmic procedure includes the workload cutoffs w_0, w_1 and w_2, as well as $\tau(w)$ and $V(w)$ for $w = \{-M + h, \ldots, w_0 - h\}$. In the next subsection, we show how to translate this output into a proposed solution.

2.7. The Proposed Policy. The mapping from a diffusion control solution to a proposed policy is less straightforward when a numerical solution is obtained than when an analytical solution is derived. More specifically, with a numerical solution there is no way to develop a proposed scheduling policy that is independent of the heavy traffic scaling parameter n: The drift of the underlying diffusion process is $\sqrt{n}(1-\rho) - s/(\tau(w) + s/\sqrt{n})$,

and a value of n must be chosen in order to compute a numerical solution to the Markov chain control problem. This quandary is dealt with in the most natural way: We set c equal to one and let $n = (1-\rho)^{-2}$. Moreover, we set the finite difference interval h equal to $1/\sqrt{n}$, so that the discretization in the Markov chain approximation procedure corresponds to individual units of inventory in the original problem. Exploratory computations revealed that the performance of the proposed policy was very insensitive to our choice of n.

Recall that we replaced the cycle length $\tau(w)$ by $\tau(w)+s/\sqrt{n}$ in the Markov chain algorithm. Therefore, in creating our proposed policy, we employ the cycle center $x_c(\tau(w)+s/\sqrt{n}, w)$. The proposed policy for the setup cost case produced product i until its inventory reached $\tilde{x}_i^c(\tilde{\tau}(\tilde{w}), \tilde{w}) + \tilde{\tau}(\tilde{w})\rho_i(1-\rho_i)/2$. In order for the expected total busy time in a cycle to be equal to $\tilde{\tau}(\tilde{w})$, the $s/\sqrt{n}$ term is not added in the second of the two terms in this expression, and we produce product i until its inventory reaches $\tilde{x}_i^c(\tilde{\tau}(\tilde{w}) + s(1-\rho), \tilde{w}) + \tilde{\tau}(\tilde{w})\rho_i(1-\rho_i)/2$.

The proposed policy is constructed just as in the setup cost problem. Define the unnormalized workload regions $\tilde{\text{I}}$, $\tilde{\text{II}}$ and $\tilde{\text{III}}$ according to whether the quantity $(1-\rho)\sum_{i=1}^N \mu_i^{-1}\tilde{I}_i(t)$ is greater than w_1, in the interval $[w_2, w_1]$ or less than w_2, respectively. Then our proposed policy is: *If* $(1-\rho)\sum_{i=1}^N \mu_i^{-1}\tilde{I}_i(t) > 0$ *then let* N *refer to the product with the smallest holding cost index* h_i*; otherwise, let* N *denote the product with the smallest value of* b_i*. The server should idle if* $(1-\rho)\sum_{i=1}^N \mu_i^{-1}\tilde{I}_i(t) > w_0$*; otherwise, if set up for product* $i < N$ *then produce this product as long as*

$$\mu_i^{-1}\tilde{I}_i(t) < \begin{cases} \frac{\left(\tau\left((1-\rho)\sum_{j=1}^N \mu_j^{-1}\tilde{I}_j(t)\right)+s(1-\rho)\right)\rho_i(1-\rho_i)}{(1-\rho)(b_i+h_i)}\left[\frac{b_i-h_i}{2}+h_N\right] & \tilde{\mathrm{I}} \\ \quad +\frac{\tau\left((1-\rho)\sum_{j=1}^N \mu_j^{-1}\tilde{I}_j(t)\right)\rho_i(1-\rho_i)}{2(1-\rho)} & \\ \frac{\left(\tau\left((1-\rho)\sum_{j=1}^N \mu_j^{-1}\tilde{I}_j(t)\right)+s(1-\rho)\right)\alpha_i\cdot\gamma_1}{1-\rho}+\alpha_i\cdot\gamma_2\sum_{j=1}^N \mu_j^{-1}\tilde{I}_j(t) & \tilde{\mathrm{II}} \\ \quad +\frac{\tau\left((1-\rho)\sum_{j=1}^N \mu_j^{-1}\tilde{I}_j(t)\right)\rho_i(1-\rho_i)}{2(1-\rho)} & \\ \frac{\left(\tau\left((1-\rho)\sum_{j=1}^N \mu_j^{-1}\tilde{I}_j(t)\right)+s(1-\rho)\right)\rho_i(1-\rho_i)}{(1-\rho)(b_i+h_i)}\left[\frac{b_i-h_i}{2}-b_N\right] & \tilde{\mathrm{III}} \\ \quad +\frac{\tau\left((1-\rho)\sum_{j=1}^N \mu_j^{-1}\tilde{I}_j(t)\right)\rho_i(1-\rho_i)}{2(1-\rho)} & \end{cases} . \qquad (74)$$

Once $\mu_i^{-1}\tilde{I}_i(t)$ reaches or exceeds this level, switch to the next product. If set up for product N, then produce this product while

$$\mu_N^{-1}\tilde{I}_N(t) < \begin{cases} \sum_{j=1}^N \mu_j^{-1}\tilde{I}_j(t)+\frac{\tau\left((1-\rho)\sum_{j=1}^N \mu_j^{-1}\tilde{I}_j(t)\right)\rho_N(1-\rho_N)}{2(1-\rho)} & \tilde{\mathrm{I}} \\ -\sum_{i=1}^{N-1}\frac{\left(\tau\left((1-\rho)\sum_{j=1}^N \mu_j^{-1}\tilde{I}_j(t)\right)+s(1-\rho)\right)\rho_i(1-\rho_i)}{(1-\rho)(b_i+h_i)}\left[\frac{b_i-h_i}{2}+h_N\right] & \\ \sum_{j=1}^N \mu_j^{-1}\tilde{I}_j-\sum_{i=1}^{N-1}\left[\frac{\left(\tau\left((1-\rho)\sum_{j=1}^N \mu_j^{-1}\tilde{I}_j(t)\right)+s(1-\rho)\right)\alpha_i\cdot\gamma_1}{1-\rho}\right. & \tilde{\mathrm{II}} \\ \left.+\alpha_i\cdot\gamma_2\sum_{j=1}^N \mu_j^{-1}\tilde{I}_j(t)\right]+\frac{\tau\left((1-\rho)\sum_{j=1}^N \mu_j^{-1}\tilde{I}_j(t)\right)\rho_N(1-\rho_N)}{2(1-\rho)} & \\ \sum_{j=1}^N \mu_j^{-1}\tilde{I}_j(t)+\frac{\tau\left((1-\rho)\sum_{j=1}^N \mu_j^{-1}\tilde{I}_j(t)\right)\rho_N(1-\rho_N)}{2(1-\rho)} & \tilde{\mathrm{III}} \\ -\sum_{i=1}^{N-1}\frac{\left(\tau\left((1-\rho)\sum_{j=1}^N \mu_j^{-1}\tilde{I}_j(t)\right)+s(1-\rho)\right)\rho_i(1-\rho_i)}{(1-\rho)(b_i+h_i)}\left[\frac{b_i-h_i}{2}-b_N\right] & \end{cases} . \qquad (75)$$

and then switch to the next product when $\mu_N^{-1}\tilde{I}_N(t)$ reaches or exceeds this level.

The solution proposed above is specified in terms of the original problem parameters, and the constants w_0, w_1, w_2 and the function $\tau(w)$ generated by the algorithmic procedure. Since $\tau(w)$ is only defined on a discrete state space, the argument of this function is rounded to the closest discrete value in the algorithmic discretization; that is, $\tau(w)$ is approximated by $\tau(y^*)$, where y^* satisfies $|y^*-w| \le |y-w|$ for all $y \in [-N+h,\ldots,w_0-h]$.

3. COMPUTATIONAL STUDY

In this section we evaluate the effectiveness of our proposed policies by conducting a series of two-product and five-product experiments for both the setup cost and setup time problems. For the two-product cases, we compare the performance of our proposed policy and two "straw" policies against a numerically derived optimal policy. A dynamic programming value iteration algorithm is used to find the optimal policy and evaluate the performance of all four policies. ¿From this data we compute the *suboptimality* for the proposed and two straw policies by

$$\text{policy's suboptimality} = \frac{\text{policy's cost} - \text{optimal cost}}{\text{optimal cost}} \times 100\% \ .$$

In implementing the value iteration algorithm (readers are referred to Markowitz for a detailed specification of the algorithm), the inventory state space was truncated to $[-150, 150]$ by $[-150, 150]$ for the setup cost cases and $[-250, 250]$ by $[-250, 250]$ for the setup time cases. To achieve three-digit accuracy of the suboptimalities, 7,000 iterations of the algorithm were required for the setup cost problem and 14,000 for the setup time problem. Due to the large number of inventory states, a dynamic programming algorithm is not feasible for the five-product cases, and thus no optimal policy is derived. Instead, discrete event simulation is used to evaluate the proposed policy and the two straw policies. To evaluate a policy for a particular scenario, we perform 5 independent runs of 6,000,000 time units for the setup cost problems and 10 independent runs of 6,000,000 time units for the setup time problems; each run starts with an empty system and statistics from the first 10,000 time units are discarded. For all scenarios in this section, we assume that the demand interarrival times, service times and setup times are exponentially distributed: service is preemptive in the two-product cases and non-preemptive in the five-product cases.

For systems with two products, we consider 20 setup cost cases and 14 setup time cases; all but two cases for each type of problem assume that the products have identical parameters. Although nearly all of our cases are symmetric, the numerical results in Reiman and Wein suggest that the heavy traffic analysis is equally accurate for symmetric

and asymmetric problems. For systems with five products, we consider six setup cost cases and four setup time cases. We focus on the two-product setting for several reasons. The optimal solution can be numerically computed in this setting, which allows us to assess the suboptimality of our proposed policies; since the optimal policy is a dynamic cyclic policy in the two-product case (i.e., the optimal policy chooses one of the three scheduling options that we allow at each point in time), we conjecture that our proposed policies are optimal in the heavy traffic limit. Also, the graphical depictions of the various policies in two dimensions (see Figures 2 through 6) help us to understand the subtleties of the behavior of this system. The two straw policies are described in §3.1, and the numerical results for the setup cost and setup time problems are given in §3.2 and §3.3, respectively. Our key observations are summarized in §3.4.

3.1 Straw Policies. To help assess the effectiveness of the proposed policy, we consider two simpler classes of cyclic policies, and use heavy traffic analysis to optimize within these classes. One is a generalized base stock policy and the other is a fixed size corridor policy similar to one considered by Sharifnia, Caramanis and Gershwin. Neither straw policy employs the $s/\sqrt{n}$ refinement that was introduced in §2.7; we discuss this issue in §3.4.

Generalized Base Stock Policy. The generalized base stock policy can be stated as follows. If the server is set up for product i, then serve this product if $\tilde{W}_i(t) < \tilde{v}_i$. If $\tilde{W}_i(t) \geq \tilde{v}_i$, then idle if product j, the next product to be produced in the cycle, has a workload level $\tilde{W}_j(t) \geq \tilde{v}_j - \tilde{y}_j$; otherwise, switch to product j at this point. Hofri and Ross (1987) prove that the make-to-order version of this policy is optimal in a two-product symmetric polling system. The generalized base stock policy can be thought of as a refined version of the cyclic base stock policy considered by Federgruen and Katalan (1993), in the sense that their policy can only insert idleness in a state-independent manner. Although the generalized base stock policy contains $2N$ parameters, the heavy traffic behavior of this policy (see Reiman and Wein for details) depends on the $\tilde{y}_i$'s only via $\max_{1\leq i\leq N} \tilde{y}_i$; let us denote this quantity by $\tilde{y}$. Hence, we set each $\tilde{y}_i$ equal to $\tilde{y}$, and optimize over the $N+1$ normalized parameters $(v_1, \ldots, v_N, y)$, where $y = \tilde{y}/\sqrt{n}$ and $v_i = \tilde{v}_i/\sqrt{n}$. In heavy traffic, this policy is equivalent to one that completes production of product i when

its inventory level reaches v_i, and employs the workload idling threshold $\sum_{i=1}^N v_i - y$; see Figure 2.

To calculate the cost associated with this policy, we move from these natural parameters to those used in §2 and §3. Under the CPR results, for a given total workload w a generalized base stock policy has cycle center

$$x_i^c(w) = v_i - \frac{\rho_i(1-\rho_i)}{\sum_{l=1}^N \rho_l(1-\rho_l)}(\sum_{j=1}^N v_j - w) \tag{76}$$

and cycle length

$$\tau(w) = 2\frac{\sum_{i=1}^N v_i - w}{\sum_{i=1}^N \rho_i(1-\rho_i)}. \tag{77}$$

Product i's average inventory cost is obtained by substituting these parameters into equation (4); i.e., $c_i(2\frac{\sum_{i=1}^N v_i - w}{\sum_{i=1}^N \rho_i(1-\rho_i)}, v_i - \frac{\rho_i(1-\rho_i)(\sum_{j=1}^N v_j - w)}{\sum_{j=1}^N \rho_j(1-\rho_j)}, w)$.

Using the CPR results, we can derive the total average cost for the generalized base stock policy for both the setup cost and time problems. In the setup cost case, total average cost is calculated by integrating the average inventory costs plus average setup costs over the stationary distribution of the normalized total workload. Since the normalized total workload W is approximated by a RBM, the total average cost is

$$\begin{aligned}\int_{-\infty}^{\sum_{i=1}^N v_i - y} \Bigg(&\sum_{i=1}^N c_i(2\frac{\sum_{i=1}^N v_i - w}{\sum_{i=1}^N \rho_i(1-\rho_i)}, v_i - \frac{\rho_i(1-\rho_i)(\sum_{j=1}^N v_j - w)}{\sum_{j=1}^N \rho_j(1-\rho_j)}, w) \\ &+ \frac{k\sum_{i=1}^N \rho_i(1-\rho_i)}{2\sum_{i=1}^N v_i - w}\Bigg) \alpha e^{-\alpha(\sum_{i=1}^N v_i - w)} dw,\end{aligned} \tag{78}$$

where $\alpha = 2\sqrt{n}(1-\rho)/\sigma^2$. For the setup time problem, the average inventory cost is similar, although the stationary distribution of W is no longer exponential, but gamma (see Coffman, Puhalskii and Reiman 1995b). The total average cost is

$$\begin{aligned}\int_{-\infty}^{\sum_{i=1}^N v_i - y} &\left(\sum_{i=1}^N c_i(2\frac{\sum_{i=1}^N v_i - w}{\sum_{i=1}^N \rho_i(1-\rho_i)}, v_i - \frac{\rho_i(1-\rho_i)(\sum_{j=1}^N v_j - w)}{\sum_{j=1}^N \rho_j(1-\rho_j)}, w)\right) \\ &\frac{\alpha(\alpha(\sum_{i=1}^N v_i - w))^\beta}{\Gamma(\beta+1)} e^{-\alpha(\sum_{i=1}^N v_i - w)} dw,\end{aligned} \tag{79}$$

where α is as above and $\beta = s\sum_{i=1}^N \rho_i(1-\rho_i)/\sigma^2$. We set n equal to $(1-\rho)^{-2}$ and use a steepest descent algorithm to find the parameters $(v_1, \ldots, v_N, y)$ that minimize (78)

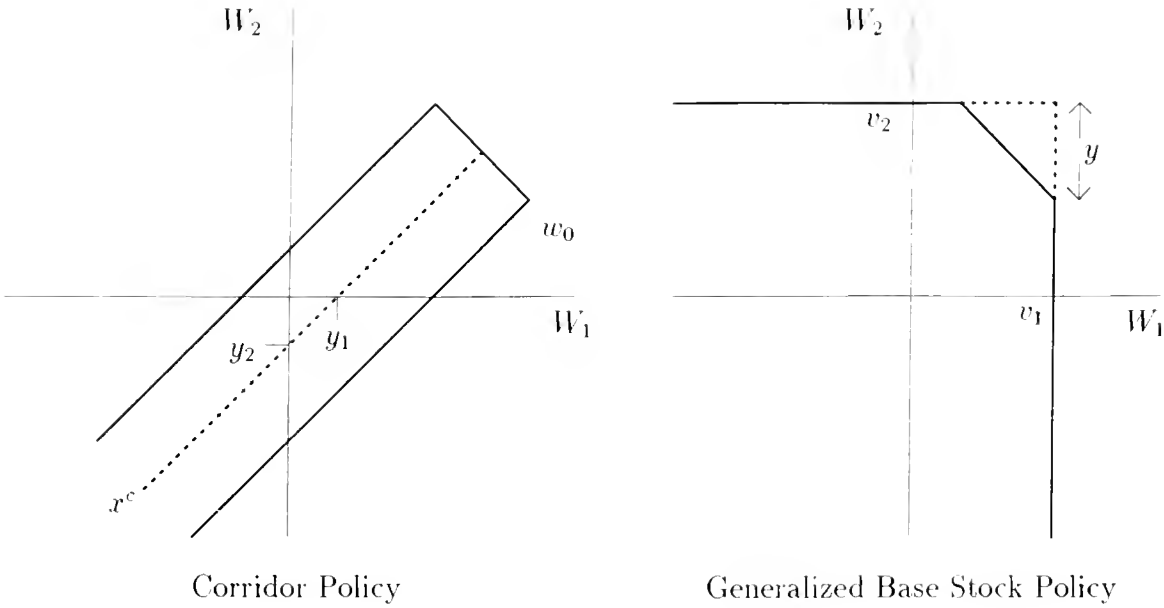

Figure 2: The two straw policies.

and (79). For both the setup cost and setup time cases, we reverse the heavy traffic scaling to obtain the proposed parameter values $\tilde{v}_i = v_i/(1-\rho)$ and $\tilde{y} = y/(1-\rho)$.

The Corridor Policy. This policy can be stated in terms of switching hyperplanes in the product workload space. The hyperplanes are created to form a fixed width corridor with its long axis orthogonal to the constant workload plane (see Figure 2). The policy represents a natural embodiment of the "constant lot size" philosophy within a dynamic stochastic framework, and is defined by $N+2$ parameters: The cycle length τ (or corridor width), the idling threshold w_0 and the parameters $(y_1, \ldots, y_N)$, which determine the intercept of the corridor's axis. We can use these variables and the notation of the previous two sections to formulate the average inventory cost of the policy. For a given workload w, the cycle center x_i^c is equal to $w/N + y_i$ and the cycle length is τ. Product i's average inventory cost for workload w is then $c_i(\tau, w/N + y_i, w)$.

Because the cycle length is independent of workload, in both the setup cost and setup time cases the diffusion process W is approximated by a RBM on $(-\infty, w_0]$, which has an exponential steady state distribution. The drifts of the RBM are c and $c - s/\tau$ for the

setup cost and time problems, respectively. Thus, the total average cost is

$$\int_{-\infty}^{w_0}\left(\sum_{i=1}^{N} c_i(\tau, \frac{w}{N} + y_i, w) + \frac{k}{\tau}\right)\frac{2\sqrt{n}(1-\rho)}{\sigma^2}e^{-\frac{2\sqrt{n}(1-\rho)}{\sigma^2}(w_0-w)}dw \tag{80}$$

for the setup cost problem, and

$$\int_{-\infty}^{w_0}\left(\sum_{i=1}^{N} c_i(\tau, \frac{w}{N} + y_i, w)\right)\frac{2(\sqrt{n}(1-\rho) - s/\tau)}{\sigma^2}e^{-\frac{2(\sqrt{n}(1-\rho)-s/\tau)}{\sigma^2}(w_0-w)}dw \tag{81}$$

for the setup time problem. The cost-minimizing parameters for (80) and (81) are determined by a steepest descent algorithm. Although we were able to use $n = (1-\rho)^{-2}$ in our computations, in the setup time problem one must be careful to choose the heavy traffic scaling factor n so that $\sqrt{n}(1-\rho)$ is greater than s/τ, thereby guaranteeing a well defined integral; this inequality is simply the stability condition that the fraction of time the server spends processing units and setting up must be less than one. Finally, the proposed parameter values are given by $\tilde{y}_i = y_i/(1-\rho)$ and $\tilde{\tau} = \tau/(1-\rho)$.

3.2. The Setup Cost Problem.

Two-Product Cases. To standardize the two-product scenarios, we set the service rates $\mu_1 = \mu_2 = 1$ and control the utilization rates ρ_i by varying the demand rates λ_i. We also set $h_2 = 1$ and, by modifying h_1, b_1 and b_2, select product 2 as the least cost product. Inventory costs and arrival rates are identical across products in the 18 *symmetric* cases, and each case is characterized by three parameters: Backorder cost, traffic intensity and setup cost per cycle. We examine all permutations of values shown in Table I; notice that some of these scenarios grossly violate the heavy traffic conditions. The parameters for the first asymmetric case are $\lambda_1 = 0.6$, $\lambda_2 = 0.3$, $h_1 = 2$, $b_1 = 10$, $b_2 = 5$ and $K = 200$. The second asymmetric case is the same as the first, except that the backorder cost is doubled to $b_1 = 20$ and $b_2 = 10$.

Table II displays the results for the 20 two-product cases and Tables III to V show the averages (for the 18 symmetric cases) over individual parameters for each policy. The switching curves for the optimal and proposed policies for the $(b = 5,\ K = 200,\ \rho = 0.9)$ two-product symmetric case is depicted in Figure 3, and corresponding curves for the

	Backorder Cost b	Setup Cost K	Traffic Intensity ρ
Low	5	20	0.5
Medium		100	0.7
High	10	200	0.9

Table I: The 18 test cases for the symmetric two-product setup cost problem.

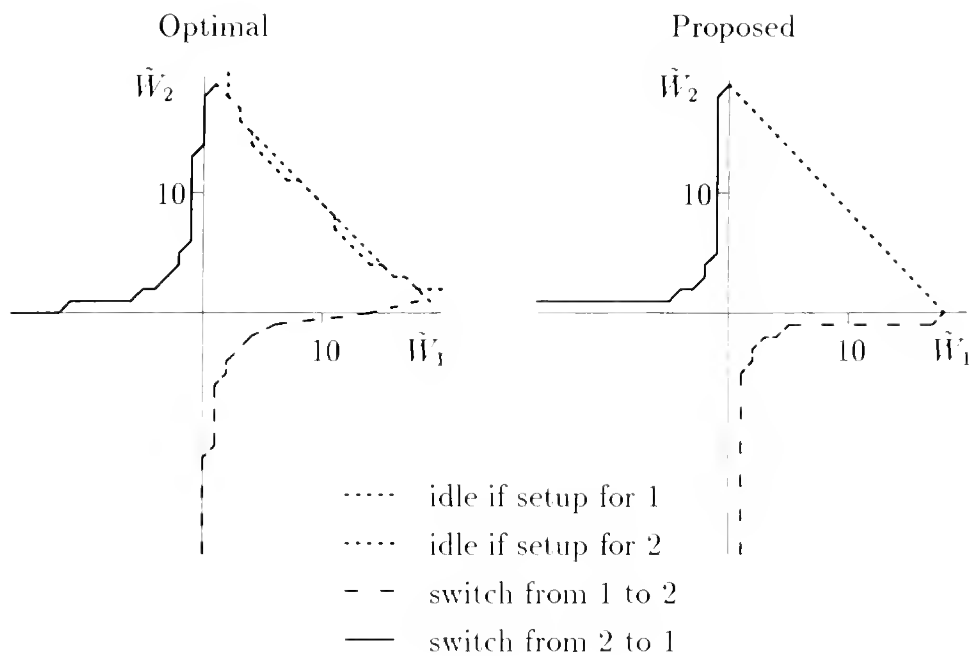

Figure 3: Switching curves for a symmetric setup cost case.

$b_1 = 10$ asymmetric case are displayed in Figure 4.

Five-Product Cases. We set $\lambda_i = 0.18$ and $\mu_i = 1$ for $i = 1, \ldots, 5$ for each of the six cases, resulting in a traffic intensity of 0.9. We also set $b_i = 5h_i$ for $i = 1, \ldots, 5$ for half the cases and $b_i = 10h_i$ for the other half. Each case is characterized by h_i, b_i and the setup cost. Four of the six cases are symmetric ($h_i = 1$ for $i = 1, \ldots, 5$) and two of the six cases are asymmetric ($h_i = i$ for $i = 1, \ldots, 5$). The average cost for each policy (along with 95% confidence intervals) is displayed in the first six rows of Table VI.

3.3 The Setup Time Problem. As in the two-product setup cost test cases, we assume that $\mu_1 = \mu_2 = 1$ and $h_2 = 1$. In the 12 symmetric scenarios, each product's

Backorder Cost b	Setup Cost K	Traffic Intensity ρ	Optimal Policy Gain	Proposed Policy	Suboptimality of Corridor Policy	Gen. Base Stock Policy
5	20	0.5	4.30	6.0%	6.5%	9.6%
5	20	0.7	6.73	8.3%	1.5%	20.1%
5	20	0.9	17.99	3.8%	7.5%	26.8%
5	100	0.5	7.00	6.2%	6.4%	18.5%
5	100	0.7	9.72	2.1%	2.6%	18.5%
5	100	0.9	20.14	1.5%	8.1%	27.4%
5	200	0.5	9.14	15.0%	7.2%	25.3%
5	200	0.7	12.24	2.6%	2.9%	21.3%
5	200	0.9	22.22	0.7%	7.5%	26.1%
10	20	0.5	5.30	14.1%	2.2%	20.0%
10	20	0.7	8.41	13.9%	4.2%	24.6%
10	20	0.9	23.58	6.0%	11.7%	33.0%
10	100	0.5	7.98	8.0%	6.1%	15.5%
10	100	0.7	11.31	4.5%	3.7%	25.5%
10	100	0.9	25.44	3.3%	9.6%	34.9%
10	200	0.5	10.21	6.4%	7.2%	18.6%
10	200	0.7	13.79	3.5%	4.7%	24.3%
10	200	0.9	27.26	2.1%	10.0%	35.0%
	Asym.	$b_1=10$	28.71	2.6%	13.1%	35.8%
	Asym.	$b_1=20$	35.76	3.4%	14.9%	45.8%

Table II: Results for the two-product setup cost cases.

inventory costs and service utilizations are identical and we vary only the backorder cost, the traffic intensity and the average setup time per cycle. Table VII reports all of the permutations of values analyzed. The first asymmetric scenario is defined by $\lambda_1 = 0.6$, $\lambda_2 = 0.3$, $\mu_1 = \mu_2 = 1$, $h_1 = 2$, $h_2 = 1$, $b_1 = 10$, $b_2 = 5$ and $s = 20$. The second asymmetric scenario is identical except that the backorder costs are $b_1 = 20$ and $b_2 = 10$.

The individual results for the 14 runs are displayed in Table VIII and policy summaries

	Backorder Cost b	Setup Cost K	Traffic Intensity ρ
Low	5.1%	8.7%	9.3%
Medium		4.3%	5.8%
High	6.9%	5.0%	2.9%

Overall Average Suboptimality = 6.0%

Table III: Average suboptimality of the proposed policy: Setup cost problem.

	Backorder Cost b	Setup Cost K	Traffic Intensity ρ
Low	5.6%	5.6%	5.9%
Medium		6.1%	3.3%
High	6.6%	6.6%	9.1%

Overall Average Suboptimality = 6.1%

Table IV: Average suboptimality of the corridor policy: Setup cost problem.

	Backorder Cost b	Setup Cost K	Traffic Intensity ρ
Low	21.5%	22.4%	17.9%
Medium		23.4%	22.4%
High	25.7%	25.1%	30.5%

Overall Average Suboptimality = 23.6%

Table V: Average suboptimality of the generalized base stock policy: Setup cost problem.

for the 12 symmetric runs are given in Tables IX to XI. In addition, Figures 5 and 6 provide a graphical depiction of the proposed and optimal policies for a symmetric case ($b = 5$, $s = 2$, $\rho = 0.9$) and the $b_1 = 10$ asymmetric case, respectively.

Results for two five-product scenarios can be found in Table VI; they are identical to the setup cost scenarios described in §3.2, except that setup times (with $s = 50$) are incurred rather than setup costs.

3.4. Observations. Our observations from the numerical results are summarized in this subsection. The five-product cases are discussed after the two-product cases.

Performance of the proposed policy: In the setup cost cases, the proposed policy's average suboptimality is 6.0% over the 18 symmetric scenarios. The policy performs very well when the heavy traffic conditions are satisfied; for example, the suboptimality is 0.7% when $b_1 = 5$, $K = 200$ and $\rho = 0.9$. Considering that the proposed policy was constructed via a heavy traffic approximation, it operates reasonably well over a wide range of system parameters, including a low utilization rate of 0.5. Not surprisingly, the policy performs worst when the traffic intensity is low, the setup costs are small, and the backorder costs are high. The policy also performs well (2.6% and 3.4% suboptimalities)

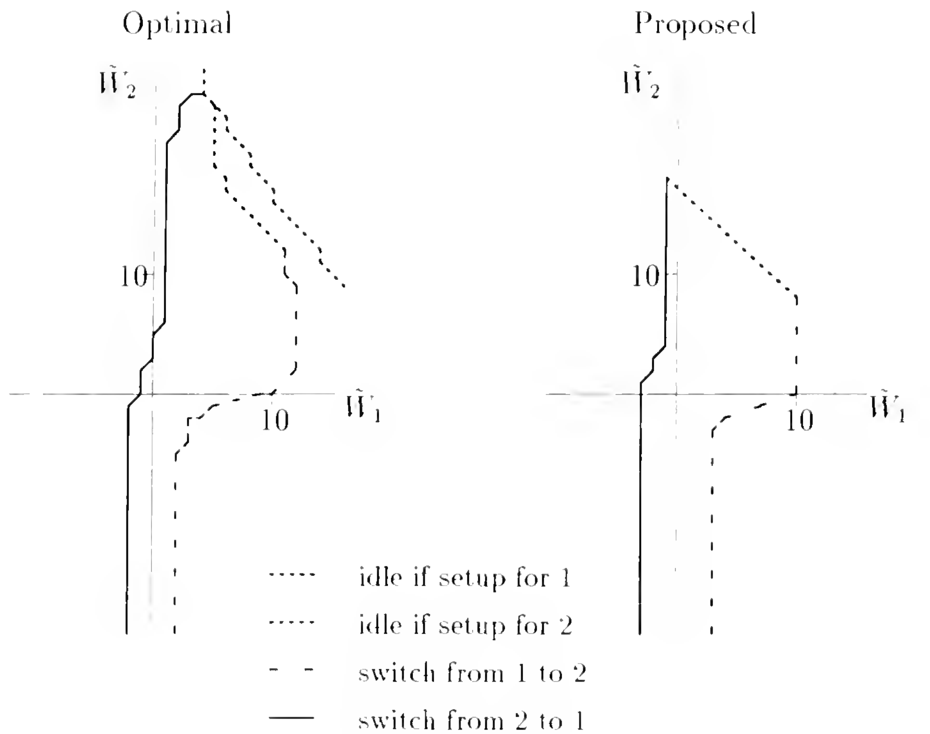

Figure 4: Switching curves for the $b_1 = 10$ asymmetric setup cost case.

in the asymmetric cases.

In the setup time cases, the average suboptimality over the 12 symmetric cases is 7.2%. The policy performs very well (1.8% average suboptimality) when the traffic intensity is high, but degrades somewhat in the lighter traffic cases. It also performs well in the asymmetric cases (1.5% and 3.3% suboptimalities).

Switching curves: The switching curves of the proposed and optimal policies are remarkably similar in Figures 3 to 6 and are unlike either the corridor or generalized base stock policies. In the two symmetric problems (Figures 3 and 5), these curves have the same general shape as predicted by our heavy traffic analysis: A distinctive constant-workload idling threshold, a wide cycle time for large positive and negative inventories and a small cycle time about the zero total workload level. In the asymmetric setup cost problem in Figure 4, the three-region categorization predicted by the heavy traffic theory is easily recognizable in the optimal policy. Figure 6 confirms that lot sizes shrink as the

Back-order Cost	Setup Cost or Time	Cost Structure	Cost of Proposed Policy	Cost of Corridor Policy	Cost of Gen. Base Stock Policy
$b = 5$	$K = 50$	Symmetric	25.32($\pm$0.46)	28.78($\pm$0.63)	33.23($\pm$0.45)
$b = 5$	$K = 500$	Symmetric	37.23($\pm$0.10)	37.25($\pm$0.32)	44.02($\pm$0.20)
$b = 10$	$K = 50$	Symmetric	36.79($\pm$0.73)	39.39($\pm$0.91)	40.54($\pm$0.68)
$b = 10$	$K = 500$	Symmetric	47.08($\pm$0.38)	46.02($\pm$0.27)	54.29($\pm$0.52)
$b = 5$	$K = 500$	Asymmetric	79.91($\pm$0.41)	86.65($\pm$0.67)	121.11($\pm$1.57)
$b = 10$	$K = 500$	Asymmetric	98.46($\pm$0.77)	105.98($\pm$1.49)	138.88($\pm$1.14)
$b = 5$	$s = 50$	Symmetric	215.4($\pm$4.9)	228.0($\pm$16.1)	214.1($\pm$2.6)
$b = 10$	$s = 50$	Symmetric	264.7($\pm$10.4)	532.5($\pm$136.8)	260.2($\pm$4.7)
$b = 5$	$s = 50$	Asymmetric	610.8($\pm$8.9)	683.9($\pm$35.2)	661.0($\pm$9.1)
$b = 10$	$s = 50$	Asymmetric	737.4($\pm$18.7)	827.9($\pm$66.5)	791.7($\pm$16.1)

Table VI: Results for the five-product cases.

	Backorder Cost b	Setup Time s	Traffic Intensity ρ
Low	5	2	0.5
Medium			0.7
High	10	20	0.9

Table VII: The 12 test cases for the symmetric setup time problem.

idling threshold is approached. Finally, as the total workload $\tilde{w}$ tends to minus infinity, lot sizes appear to be growing roughly with $-\tilde{w}$ in Figure 5 and with $\sqrt{-\tilde{w}}$ in Figure 6.

Two key differences between the proposed and optimal policies emerge from studying Figures 3 to 6; numerical results (not reported here) verify that both discrepancies dissipate as the traffic intensity approaches unity, and get more severe in the lower utilization cases. First, in all four figures, the proposed heavy traffic policies have a tendency to backorder more than the optimal policy; this observation is most obvious in the upper right portion of Figure 5. Because CPR's time scale decomposition does not hold precisely for the original stochastic system, the optimal policy hedges against backorders slightly more than the proposed heavy traffic policy, which assumes that the inventory levels respond in a deterministic fashion in the fluid limit. In terms of these figures, the cycle lengths (i.e., the distance along the total workload line between the solid and dashed curves) tend to be slightly smaller in the optimal policy; consequently, the workload pro-

Backorder Cost b	Setup Time s	Traffic Intensity ρ	Optimal Policy Gain	Proposed Policy	Suboptimality of Corridor Policy	Gen. Base Stock Policy
5	2	0.5	3.84	9.0%	12.8%	6.6%
5	2	0.7	7.45	7.4%	8.3%	7.8%
5	2	0.9	25.31	1.4%	6.7%	9.5%
5	20	0.5	13.58	10.1%	24.3%	19.9%
5	20	0.7	26.39	5.9%	16.0%	9.6%
5	20	0.9	79.40	1.4%	8.7%	3.5%
10	2	0.5	5.15	9.3%	11.0%	11.8%
10	2	0.7	9.85	8.2%	7.1%	9.0%
10	2	0.9	33.24	1.8%	4.5%	13.7%
10	20	0.5	17.75	18.7%	26.5%	17.9%
10	20	0.7	33.67	10.7%	16.8%	12.8%
10	20	0.9	98.93	2.6%	7.0%	5.9%
	Asym.	$b_1=10$	104.91	1.5%	30.6%	11.5%
	Asym.	$b_1=20$	129.42	3.3%	60.9%	15.8%

Table VIII: Results for the two-product setup time cases.

	Backorder Cost b	Setup Time s	Traffic Intensity ρ
Low	5.9%	6.2%	11.8%
Medium			8.1%
High	8.5%	8.2%	1.8%

Overall Average Suboptimality = 7.2%

Table IX: Average suboptimality of the proposed policy: Setup time problem.

cess spends less time in the backorder region and some of our remarks in §2.5 regarding the inventory levels at switching epochs only hold in very heavy traffic. This limitation of the heavy traffic theory was also noted in Wein.

The other main discrepancy occurs near the idling threshold in the asymmetric cases: In Figure 4, the optimal lot sizes for product 1 decrease, rather than stay constant, as the workload idling threshold is approached, and in Figures 4 and 6 there is a different idling threshold for each product. This discrepancy can be explained as follows: When the total workload is positive, the switchover cost (in Figure 4) or time (in Figure 6) makes it beneficial to be set up for product 1, so as to efficiently protect against costly product 1 backorders. If ρ is not close to one, then it is likely that the total inventory workload

	Backorder Cost b	Setup Time s	Traffic Intensity ρ
Low	12.8%	8.4%	18.7%
Medium			12.1%
High	12.1%	16.5%	6.7%

Overall Average Suboptimality = 12.5%

Table X: Average suboptimality of the corridor policy: Setup time problem.

	Backorder Cost b	Setup Time s	Traffic Intensity ρ
Low	9.5%	9.8%	14.0%
Medium			9.8%
High	11.8%	11.6%	8.2%

Overall Average Suboptimality = 10.7%

Table XI: Average suboptimality of the generalized base stock policy: Setup time problem.

will increase while producing product 2; that is, the increase in product 2's inventory workload will exceed the reduction in product 1's inventory workload. The optimal policy takes advantage of this imbalance by allowing product 2's inventory to grow beyond the product 1 idling threshold; this extra product 2 inventory allows the server to idle while setup for product 1.

Performance of the corridor policy. The corridor policy exhibits erratic behavior. The policy performs very well in the symmetric setup cost cases (it outperforms the proposed policy in six of the 18 scenarios, all of which have low or medium utilizations), but degrades slightly at high utilization. A comparison of Figures 2 and 3 leads us to believe that the parameters of the policy are being set correctly at high utilizations, and the performance degradation is due to the corridor policy's inability to employ mean lot sizes that are state-dependent.

The corridor policy does not perform as well in the symmetric setup time cases; it is not able to increase the cycle length τ for small total workloads and so has difficulty recovering from this high backorder region. In contrast to the symmetric setup cost cases, the corridor policy's performance diminishes in light traffic; we have not determined how

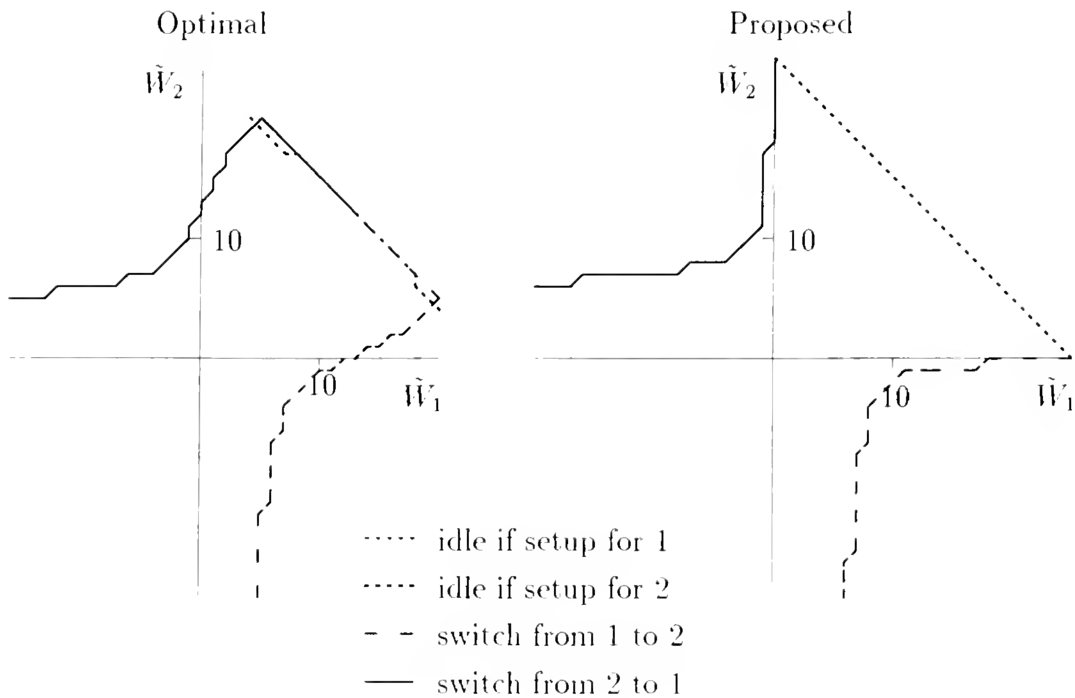

Figure 5: Switching curves for a symmetric setup time case.

much of this degradation is due to the inaccuracy of the heavy traffic approximation at low utilizations, and how much is intrinsic to the policy.

The corridor policy performs much worse when asymmetry is present: Its suboptimality is 13.1% and 14.9% in the setup cost cases and increases to 30.6% and 60.9% in the setup time cases. Comparing Figures 2, 4 and 6, it would appear that the corridor policy would never be very close to optimal for an asymmetric problem. In fact, Figure 6 suggests that a *hyperplane corridor* policy (see Figure 7 of Sharifnia, Caramanis and Gershwin) might perform reasonably well in the asymmetric setup time problem; in the two-product case, the two lines forming the corridor in Figure 2 would not be parallel in the hyperplane corridor policy, but would intersect at an idling point in the upper right portion of the graph and generate a cone-shaped corridor emanating out in the southwesterly direction.

Performance of the generalized base stock policy. The generalized base stock policy performs better in the setup time cases than in the setup cost cases: Its average suboptim-

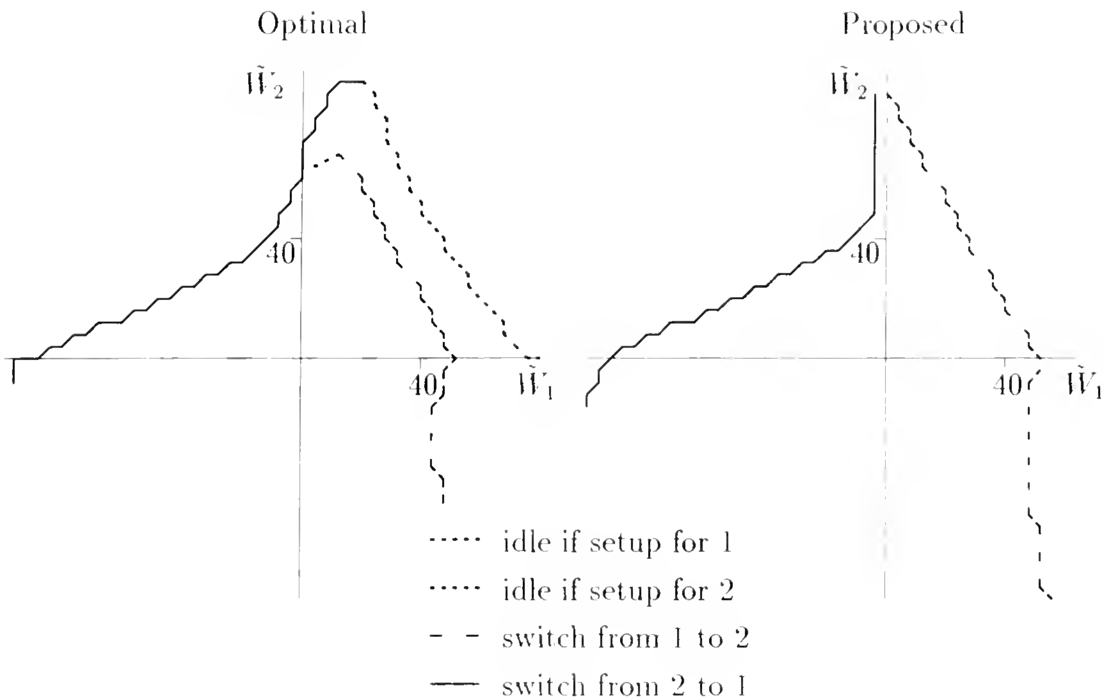

Figure 6: Switching curves for the $b_1 = 10$ asymmetric setup time case.

ality is 23.6% for the 18 symmetric setup cost scenarios and 10.7% for the 12 symmetric setup time cases. In contrast to the corridor policy, the generalized base stock policy's use of large lot sizes when the total workload is negative is a key reason for its ability to avoid poor performance in the symmetric setup time cases; however, these large lot sizes lead to considerable backordering in the setup cost scenarios. Like the corridor policy, the generalized base stock policy's performance deteriorates at high utilizations in the setup cost cases and at low utilizations in the setup time cases.

The generalized base stock policy's suboptimality is 35.8% and 45.8% in the asymmetric setup cost cases and 11.5% and 15.8% in the asymmetric setup time case. It is interesting to note that the generalized base stock policy handles expensive inventory in a manner opposite to that of the proposed policy. The order-up-to level of the most costly good is set larger than those of less expensive products to reduce the risk of expensive backordering; in contrast, the proposed policy minimizes the excess or deficit amounts of expensive inventory. It is clear that the generalized base stock policy is incapable of

closely approximating the optimal setup cost solution in Figure 4.

Five-product examples: In the six setup cost cases in Table VI, the proposed and corridor policies are roughly comparable in the two $K = 500$ symmetric cases and the corridor policy is about 10% more costly in the two $K = 50$ symmetric cases and about 8% more expensive in the two asymmetric cases. The generalized base stock policy does not fare as well in the four symmetric setup cost cases, incurring an 18.7% cost increase relative to the proposed policy, on average. Once again, the generalized base stock policy performs very poorly in the asymmetric setup cost cases.

In contrast, the generalized base stock policy performs slightly better than the proposed policy in the two symmetric setup time cases in Table VI, and is about 8% more costly than the proposed policy in the two asymmetric cases. Both of these policies outperform the corridor policy in the four setup time cases. The corridor policy is about 12% more costly than the proposed policy in the two asymmetric cases, but performs extremely poorly in one of the two symmetric cases.

To compare the relative cost differences in the two-product cases and the five-product cases, we can identify the six symmetric cases in Table VI with their two-product counterparts in Tables II and VIII; for example, the first scenario in Table VI corresponds to the $b = 5$, $K = 20$, $\rho = 0.9$ case in Table II. For the four setup cost cases, the cost increases of the straw policies relative to the proposed policy are somewhat larger for the two-product cases: the generalized base stock policy's average cost increase is 5.8% for the two-product cases versus 4.7% for the five-product cases, and the corresponding quantities for the generalized base stock policy are 26.3% and 18.7%, respectively. For the two setup time cases, the average cost increase of the generalized base stock policy is 2.7% for the two-product scenarios and -1.1% for the five-product cases. Disregarding the poor performance of the corridor policy in one of the five-product symmetric setup cost scenarios, it appears that the relative cost advantage of the proposed policy degrades slightly when the number of products increases from two to five; however, further experiments are required to fully investigate this issue.

Lack of Robustness. Simulation results not reported here show that the performance of the three policies are rather sensitive to the policy parameters, particularly in the setup

time problem; this is somewhat surprising, given the robustness of some simpler models (e.g., the EOQ model) that capture the tradeoff between inventory costs and setups. Because it is unable to increase its lot sizes as the total inventory decreases, the corridor policy is clearly the least robust of the three policies: if the corridor width is set too narrow (as apparently happened in the eighth row of Table VI) then stability problems can set in (notice the confidence intervals for this case).

The $s/\sqrt{n}$ Refinement. Recall that the $s/\sqrt{n}$ refinement described in §2.7 is incorporated into the proposed policy, but not the two straw policies. We tested all three policies with and without the refinement, and summarize our findings here. The refinement had a minor effect on the performance of the proposed policy in the $\rho = 0.9$ cases; however, by decreasing the cycle length, it significantly improved performance in the lower utilization cases. The refinement had a mixed influence on the generalized base stock policy, sometimes improving and sometimes degrading performance; overall, it slightly impaired performance. The refinement had a negative effect on the corridor policy, and let to a severe stability problem in the eighth row of Table VI.

Summary. Although additional asymmetric cases need to be investigated before drawing definitive conclusions for the two-product problems, our observations can be summarized as follows.

The proposed policy performs very well in the 34 two-product cases: Figures 3 to 6 confirm that it captures nearly all of the complexities of the optimal policy, its suboptimality is 6.0% over the 34 cases (and 2.1% over the 12 cases that do not obviously violate the heavy traffic conditions), and it is quite robust with respect to the heavy traffic conditions, especially considering that most potential industrial applications for the SELSP are in settings with high traffic intensities. However, the relative superiority of the proposed policy appears to degrade slightly as the number of products increases, and this issue requires further investigation.

The two straw policies are not flexible enough to consistently capture the subtleties of the optimal policy. The corridor policy outperforms the generalized base stock policy in 24 of the 34 two-product examples, and its average suboptimality is 11.2% as compared to 19.5% for the generalized base stock policy. Nonetheless, in the setup time cases the

corridor policy fails to use large lot sizes when the total workload is negative and can perform erratically (see Table VIII). The generalized base stock policy is never close to optimal and performs poorly in the asymmetric setup cost cases.

4. CONCLUDING REMARKS

From a practical standpoint, the stochastic economic lot scheduling problem is arguably the most important scheduling problem in operations management. Unfortunately, there has been no success in obtaining an optimal solution to this problem using standard techniques (e.g., semi-Markov decision process theory). In this paper we restrict ourselves to the class of dynamic cyclic policies, where the server has three options at each point in time: Idle, produce the product that is currently set up, or switch over to the next product in the fixed sequence. Using heavy traffic analysis, in particular the averaging principles derived by Coffman, Puhalskii and Reiman, we make considerable progress on this problem: For the setup cost problem, the optimal heavy traffic lot-sizing policy is derived in closed form, and the idleness policy is reduced to the numerical calculation of a single threshold value. For the setup time problem, some key qualitative characteristics of the optimal heavy traffic policy are derived; moreover, regardless of the number of products, we reduce the problem to a one-dimensional diffusion control problem that is solved numerically.

The explicitness of our results, coupled with the surprisingly intricate behavior of the optimal policy, leads to a number of new insights into the optimal solution to the SELSP. These insights are summarized in §1.8 and §2.5, and describe how the dynamic lot-sizing policy depends upon whether (i) setup costs or setup times are incurred, (ii) the cost structure is symmetric or asymmetric across products and (iii) the total workload embodied in the current finished goods inventory is much less than zero, in a neighborhood containing zero, or larger than zero and near the optimal idling threshold.

We also perform a heavy traffic analysis of two classes of policies that are closely related to ones analyzed by Federgruen and Katalan (1993) and Sharifnia, Caramanis and Gershwin; the unified treatment of the optimal policy and the two straw policies (see, in particular, Figures 2 through 6) makes transparent the relative strengths and

weaknesses of the straw policies. A computational study is undertaken to compare the proposed policy and these two straw policies to the numerically computed optimal policy in 34 two-product examples. The computational study (see §3.4 for a description of the key observations) confirms that the insights summarized in §1.8 and §2.5 do indeed occur in the optimal policy. Moreover, numerical results for the two-product examples show that the proposed policy is reasonably close to optimal, is robust with respect to the heavy traffic conditions, and outperforms the two straw policies; in contrast, the two straw policies lack the sophistication required to imitate the subtleties of the optimal policy, and their behavior is somewhat erratic.

The computational study also includes 10 five-product examples, and the results suggest that the relative superiority of the proposed policy degrades slightly as the number of products increases. In the last decade, researchers have improved the accuracy and robustness of heavy traffic approximations through the use of various heuristic refinements (e.g., Harrison 1988, Nguyen 1995, Lennon 1995). Although we have investigated one such refinement (see §2.7), we believe that other refinements to the proposed policy may lead to significant improvements and should be investigated; such refinements may be necessary in order to develop effective policies for problems with many products and moderate traffic intensities.

This study has only considered dynamic cyclic policies, where each class is served once per cycle in a fixed sequence. More generally, the cost of dynamic *periodic* policies (or *polling tables*), where each product can be produced more than once in a cycle, can be evaluated using the methods developed here. However, we have not yet found an efficient method for optimizing within this class of policies, and a thorough investigation of dynamic periodic and nonperiodic policies is left for future research.

ACKNOWLEDGMENT

This research was supported by a grant from the Leaders for Manufacturing Program at MIT, NSF grant DDM-9057297 and EPSRC grant GR/J71786. The last author thanks the Statistical Laboratory at the University of Cambridge for its hospitality while some of this work was carried out.

References

Anupindi, R. and S. Tayur. 1994. Managing Stochastic Multi-Product Systems: Model, Measures, and Analysis. Graduate School of Industrial Administration, Carnegie Mellon University, Pittsburgh, PA.

Beneš, V. E., L. A. Shepp, and H. S. Witsenhausen. 1980. Some Solvable Stochastic Control Problems. *Stochastics* **4**, 39-83.

Bourland, K. E. and C. A. Yano. 1995. The Strategic Use of Capacity Slack in the Economic Lot Scheduling Problem with Random Demand. *Management Science* **40**, 1690-1704.

Coffman, E. G., Jr., A. A. Puhalskii, and M. I. Reiman. 1995a. Polling Systems with Zero Switchover Times: A Heavy-Traffic Averaging Principle. *Annals of Applied Probability* **5**, to appear.

Coffman, E. G., Jr., A. A. Puhalskii, and M. I. Reiman. 1995b. Polling Systems in Heavy-Traffic: A Bessel Process Limit. In preparation.

Cooper, R. B., S.-C. Niu, and M. M. Srinivasan. 1992. A Decomposition Theorem for Polling Models: The Switchover Times are Effectively Additive. *Operation Research*, to appear.

Cox, D. R. and W. L. Smith. 1961. *Queues*. Methunen, London.

Dobson, G. 1987. The Economic Lot Scheduling Problem - Achieving Feasibility Using Time-Varying Lot Sizes. *Operations Research* **35**, 764-771.

Elmaghraby, S. E. 1978. The Economic Lot Scheduling Problem (ELSP): Review and Extensions. *Management Science* **24**, 587-598.

Federgruen, A. and Z. Katalan. 1993. The Stochastic Economic Lot Scheduling Problem: Cyclic Base-Stock Policies with Idle Times. Graduate School of Business, Columbia University, New York, NY.

Federgruen, A. and Z. Katalan. 1994. Customer Waiting Time Distributions under Base-Stock Policies in Single Facility Multi-Item Production Systems. Graduate School of Business, Columbia University, New York, NY.

Fuhrmann, S. W. 1992. A Decomposition Result for a Class of Polling Models. *Queueing Systems* **11**, 109–120.

Gallego, G. 1990. Scheduling the Production of Several Items with Random Demands in a Single Facility. *Management Science* **36**, 1579-1592.

Graves, S. C. 1980. The Multi-Product Production Cycling Problem. *AIIE Transactions* **12**, 233-240.

Harrison, J. M. 1985. *Brownian Motion and Stochastic Flow Systems.* John Wiley, New York.

Harrison, J.M. 1988. Brownian Models of Queueing Networks with Heterogeneous Customer Populations. In *Stochastic Differential Systems, Stochastic Control Theory and Applications*, IMA Volume 10, W. Fleming and P.L. Lions (eds.). Springer-Verlag, New York, 147-186.

Hofri, M. and K. W. Ross. 1987. On the Optimal Control of Two Queues with Server Set-up Times and its Analysis. *SIAM Journal of Computing* **16**, 399-420.

Hsu, W. 1985. On the General Feasibility Test of Scheduling Lot Sizes for Several Products on One Machine. *Management Science* **29**, 93-105.

Iglehart, D. L. and W. Whitt. 1970. Multiple Channel Queues in Heavy Traffic. I and II. *Advances in Applied Probability* **2**, 150–177 and 355–364.

Kushner, H. J. 1977. *Probability Methods for Approximations in Stochastic Control and for Elliptic Equations.* Academic Press, New York, NY.

Kushner, H. J. and P. G. Dupuis. 1992. *Numerical Methods for Stochastic Control Problems in Continuous Time.* Springer-Verlag, New York, NY.

Leachman, R. C. and A. Gascon. 1988. A Heuristic Scheduling Policy for Multi-Item, Single Machine Production Systems with Time-Varying Stochastic Demands. *Management Science* **34**, 377-390.

Lennon, T. 1995. Response-Time Approximations for Multi-Server, Multi-Class Production Systems with Significant Setups. Department of Industrial and Operations Engineering, University of Michigan, Ann Arbor, MI.

Mandl, P. 1968. *Analytical Treatment of One-Dimensional Markov Processes.* Springer-Verlag, New York, NY.

Markowitz, D. M. 1995. Heavy Traffic Analysis of Scheduling Problems for Multiclass Queues with Setups. PhD thesis, Operations Research Center, MIT, Cambridge, MA, in preparation.

Morse, P. 1958. *Queues, Inventories and Maintenance.* John Wiley, New York.

Nguyen, V. 1995. Fluid and Diffusion Approximations of a Two-station Mixed Queueing Network. *Mathematics of Operations Research* **20**, 321-354.

Qiu, J. and R. Loulou. 1995. Multiproduct Production/Inventory Control under Random Demands. *IEEE Transactions on Automatic Control* **40**, 350-356.

Reiman, M. I. 1984. Open Queueing Networks in Heavy Traffic. *Math. of Operations Research*, **9**, 441-458.

Reiman, M. I. 1988. A Multiclass Feedback Queue in Heavy Traffic. *Advances in Applied Probability* **20** , 179–207.

Reiman, M. I, and Wein, L. M. 1994. Dynamic Scheduling of a Two-Class Queue with Setups. Submitted for publication.

Sharifnia, A., M. Caramanis and S. B. Gershwin. 1991. Dynamic Setup Scheduling and Flow Control in Manufacturing Systems. *Discrete Event Dynamic Systems: Theory and Applications* **1**, 149-175.

Sox, C. R. and J. A. Muckstadt. 1995. Optimization-Based Planning for the Stochastic Lot Scheduling Problem. Operations Research and Industrial Engineering Dept., Cornell U., Ithaca, NY.

Wein, L. M. 1992. Dynamic Scheduling of a Multiclass Make-to-Stock Queue. *Operations Research* **40**, 724-735.

Zipkin, P. 1991. Computing Optimal Lot Sizes in the Economic Lot Scheduling Problem. *Operations Research* **39**, 56-63.